QL Today

Volume 5
Issue 4
Nov/December
2000

ISSN 1432-5454

The Magazine about QL, QDOS, Sinclair Computers, SMSQ...

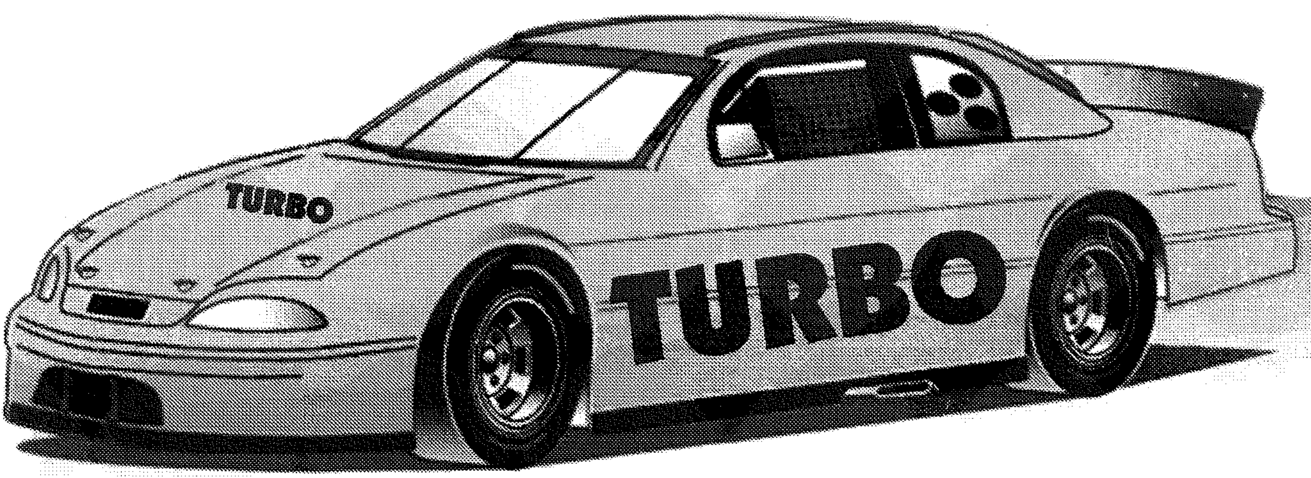
3 Ways To Speed Up Your BASIC Programs:

Update Your Hardware:
Q40

OR

Update Your Software:
QPC2
Version 2

and/or ...



... get the new TURBO for YOUR system!

Contents

Advertisers

in alphabetical order

QL *Today*

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
Mobile +49 1701 222277
email: JMerz@j-m-s.com
email: QLToday@j-m-s.com

English office:

Q Branch
20 Locks Hill
Portslade
BN41 2LB
United Kingdom

Tel. +44 1273 386030
Mobile +44 7836 745501
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk
email: QLToday@j-m-s.com

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@dj.softnet.co.uk
email: QLToday@J-M-S.com

Co-Editor:

Bruce Nicholls
57 Shaftesbury Rd
Romford
Essex RM1 2QJ
United Kingdom

Tel +44 1708 755759
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: QLToday@j-m-s.com

QL *Today* is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL *Today*** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April	Issue 2: 30 June
Issue 3: 30 August	Issue 4: 30 October
Issue 5: 30 December	Issue 6: 28 February

QL *Today* reserves the right to publish or not publish any material submitted. Under no circumstances will **QL *Today*** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL *Today***. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

Editorial

Dilwyn Jones

Thanks very much to Bruce Nicholls for keeping this seat warm while I went through my big event in September. Now my turn to keep it warm, as Bruce is having a big event of his own - changing jobs! Best of luck, Bruce.

QL2000 was another big event for all of us who attended. And I mean all of us, visitors came from France, Switzerland, Germany, America and The Netherlands to make this a truly international event. New products were demonstrated all over the place - from Jonathan Dent's appearance with a small group of Swiss QL users with his soql TCP/IP stack software, for which Al Boehm publicly presented him with a cash award from American QL users for his work on soql, to Simon Goodwin linking a digital camera to a QL and transferring pictures back and forth, to demonstrations of software-only QL midi and the QL Sampled Sound System, a range of CD-ROMs for QL emulator systems from Q-Celt and the entire software library on CD from Quanta at last! QPC2 version 2 (albeit an early version) was finally on sale, complete with colour drivers. And the memorable long wait for the saturday night meal, probably a record in itself! An in-depth discussion on the future of the QL took place on the sunday, replacing what had at first been billed as discussion on the future direction of Quanta, but the chairman said that the possibility of Quanta becoming a multi-platform organisation now seemed dead in the water, so the discussion focused on how the future should be tackled and a number of interesting ideas were discussed, including the production of a CD-ROM of QL emulators freely distributable within copyright restrictions aimed at trying to attract former QL users back to the fold.

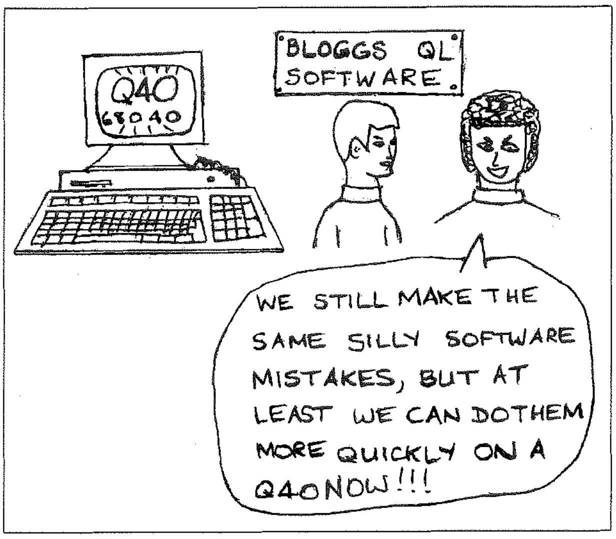
Jonathan Dent bringing soql to fruition is one thing; it also needs software. This is where Jonathan Hudson steps in, as the author who has the experience behind him of porting browsers, email and ftp programs to run on the TCP/IP system for uQLx (the QL-Socket system). Jonathans Dent and Hudson are now co-operating on the integration of the software and a project to establish an industry standard TCP/IP stack system has been established on sourceforge.net. The QL is already blessed with an established strong presence on the internet, so here's hoping we can press forward quickly. Jonathan Hudson reckoned at QL2000 that this ought to bear fruit by the new year.

Nasta's enthusiasm for the Goldfire project seems undiminished - he has been publishing diagrams on the web and so on. While the QL scene is well served by emulators and to some extent by the Q40 (although supply problems have dogged that to some extent), we are still currently lacking an expansion unit for existing QL/Aurora hardware now that the Super Gold Card is not in production.
It's that time of year when we start to think of our wish-lists. My main one has already arrived - the colour drivers version of QPC. My next wish would be to be able to hook a CD-ROM drive to Qubide and read the same CDs that I can from QPC (i.e. a QXLWIN style driver for Qubide CD-ROMs and hard disks for universal QL compatibility). A better sound system would be nice too - develop from the Q40 and QDOS Classic sound systems, and possibly add file compression to reduce the space taken by the sound files. And finally, on the software front, I'd like a graphics program to take advantage of the colour drivers, complete with colour printer drivers and ability to handle Web type files such as JPEG, GIF and PNG (some of this already possible as PROGS have been working on ProWesS) and software to more easily create Web pages and manage my web site from the QL. That is, less use of my PC, more of QDOSMSQ!

I don't ask for much do I!!!

Happy QLing-new-year to you all.

Cartoon


Q40
68040
BLOGGS QL SOFTWARE
WE STILL MAKE THE SAME SILLY SOFTWARE MISTAKES, BUT AT LEAST WE CAN DOTHEM MORE QUICKLY ON A Q40NOW!!!

NEWS

Just Words!

Solvit-Plus now Freeware

SOLVIT-PLUS is now freeware and is available from Just Words! or the main public domain libraries. The program comes on 1 HD disk that contains both the pointer and non-pointer versions of the program. It can also be downloaded from the Just Words! web page.

New 'Poundware' Range

SOLVIT-PLUS is the flagship of the Just Words! 'poundware' range of public domain or freeware programs and material. The poundware range is supplied on HD disks . In addition to SOLVIT-PLUS the range currently includes:

SPELLING-CRIB: The popular program giving help with difficult words. Comes with a 65,000 word QTYP UK English dictionary.

SPELLING-CRIB (USA): Comes with a 65,000 QTYP USA English dictionary. The dictionary is also in DP Spellchecker format.

250,000+ WORDS: Over a quarter of a million English words in a compressed file. Suitable for importing into SOLVIT-PLUS.

WORD LISTS: Word lists suitable for importing into SOLVIT-PLUS or as the basis for a spell checking dictionary. Disk 1 has Danish, French, Italian, Norwegian, Spanish and Swedish. Disk 2 has Dutch, German, UK English and USA English.

QTYP DICTIONARIES: Dutch, French, Spanish, UK English, USA English. Also contains a routine for expanding QTYP dictionaries into text files.

LANGUAGE DISKS: Currently available in Dutch, French and Spanish. Each disk contains word lists and QTYP dictionaries plus a version of Spelling Crib configured for the language.

RWAP

QL Cosmos is now at v2.04 - this version has been altered slightly to ensure that it will run on JS ROM QLs and earlier.

We have now released v1.02 of the ***ProWess ESC/P2*** drivers. This version incorporates an improved 720dpi mode and also a quicker monochrome mode. The code has also been amended to ensure that the paper is always ejected at the end of a page(this was a problem if you did not set the page size correctly).

Flightdeck is now v1.05. This version allows for longer delay timers to be set for use on the Q40 and other fast QL systems. This version also overcomes problems with running the program without the Pointer Environment being present.

We have also now released a series of smaller maps of the British Isles for Q-route users. These have been cut out of BIG Britain map and allow for more detail than their larger cousin. Maps exist for the following areas: Scotland, NE England, NW England, South & West Yorkshire, Wales & Derbyshire, South Britain and the London area. Each Map costs £2.

RWAP is also having a January Sale - order 2 or more items and receive the cheapest item at half price, the offer ends 31/01/2001.

NESQLUG

(New England Sinclair QL Users Group) is moving its main web site to

http://www.geocities.com/nesqlug/

and this site will link to sites of other NESQLUG members. The Web Master will be Rigel Cable. Please direct any questions or comments to Rigel at **cable@cyberportal.net** This site will be under construction for a while. It is our intent to make it a QL friendly site that will work fine with the lynx browser.

Nasta News

I've just created a new e-group at egroups.com. It is called QLhardware (surprise, surprise!). It is intended to contain specs, discussions, rants, news about the hardware of and for the QL computer and it's successors. The group is set up so that subscribers can post and upload and download files. There is also a web-accessible message archive for subscribers. This and the file area will grow rather slowly as I am pressed for time, but I hope that people interested in the topic of the group will subscribe and contribute. There are a number of different files I am willing to part with, amongst other things schematics and drawings of Qubide, Aurora, etc. Also, I will keep everyone posted on the status of GoldFire.

Here are the access instructions:

Web access:

http://www.egroups.com/group/QLhardware

You can subscribe and set your subscription settings. The messages can be mailed to you, or a digest can be mailed to you, or you can opt to view the messages on the web only.

Email access:

Post message: **QLhardware@egroups.com**

Subscribe: **QLhardware-subscribe@egroups.com**

(no message body is required)

Unsubscribe: **QLhardware-unsubscribe@egroups.com** (no message body is required)
Subscription is subject to moderators discretion, i.e. mine :-) so it might take a day till access is granted.

TF Services

I no longer accept credit cards. All UK banks now require electronic transfer. This means
a) I pay £600 for a machine for 3 years use and no prices for upgrades or
b) I pay £15 plus per month rental for a machine.
ie Banks are putting up two fingers to small traders.
If you find this as disagreeable as I do, then please lobby someone - ie your own bank, your MP etc.

Jérôme Grimbert

Just a quick news: the sprite editor is finally working also on QPC2 v2 (the one with high colour in it).
I finally get around the sprite caching facility in GD2, so even if it is not perfect (sometime, some colors are not updated right when requested) at least now it is usable. I should update it again (later) to enhance the readability of unavailable item (all the Q40 native mode related one).
When used on a non-Q40, you cannot edit the Q40 mode. (well, tell me how to display it accurately?) available as usual via
http://grimbert.cjb.net/
P.S.: I should also make some manual for the hundred buttons!

Quanta

We are pleased to announce a major upgrade to the original QL Users Email Database now found on **www.quanta.uni.cc**. New features include much cleaner and simplified access, and a communications facility. The database at the time of writing, has some 388 entries, with numbers steadily increasing. Access to database functions is now achieved directly from Quanta's homepage (which incidentally has also been completely re-hashed!). For instance, if you have forgotten your password or simply do not have one, all you do is enter your email address on Quanta's homepage, select the forgot/none option, and press Go!
The communications facility has been introduced to keep up with inter-computer voice and chat communications. Two options are available. Both use a pop-up window thus allowing you to continue surfing the net. The Comms option displays details of other users logged on, giving details such as their IP address and software package/ID employed for direct voice communications. The QL Users Chat option additionally displays text input and output windows.
Since it would be ideal to know when other users might be logged on, you can place a diary entry in your database record and this will be displayed on Quanta's homepage.

Quanta Links Page

This has been removed due to misuse by advertising organisations on the net. A new links list is now generated however from your database URL and site name entries. If you placed URL details on the old links page, these will no longer be visible, so update your database entry.

More on database entries

If you have not yet updated your entry since the databases relocation to **www.quanta.uni.cc,** please do so. If you do not have a password simply go to www.quanta.uni.cc and get one. Make sure you enter the email address this letter has been sent to! Some optional database entries include for instance, Quanta and QL Today membership. Completing these entries are quite important because not only will these details be used when sending information to different interest groups, some future facilities will only be available to certain interest groups.

Thierry Godefroy

A qdos-gcc re-compiled version of ProWesS is available in the download section of my Web site:
http://qdos.cjb.net/english/download.html
The speed gain is sensible although not impressive (about 30%).
qdos-gcc is now available from my Web site together with a howto...
http://qdos.cjb.net/english/qdosgcc.html

PROGS

Thanks to the help of Marcel Kilgus, I have just put a new version of ProWesS on my ftp site
ftp://ftp.triathlon98.com/pub/ProWesS/
The changes are (see also version.txt file):
- screen driver qlso supports QPC/QXL high colour mode
- 16 bit bitmap driver (used by QPC/QXL and Q40 high colour drivers) bugfix (was a problem when displaying fonts).
- some problems fixed when booting. These were caused by a minor bug in the wait program. With this bugfix, the "&wait -wait 300" and "&waitProWesS" lines can be removed from the startup file.

ERGON Freeware

Davide Santachiara has added to the list of QL software formerly sold by Ergon Development in Italy which has now been made freeware and available for download from his web site, complete with English and Italian manuals in Quill format:

http://www.geocities.com/dsantachiara

Floppy Disk Utilities

Especially designed for the (Super) Gold Card user. This program fully supports DD, HD (1.44 Mb) and ED disks (3.2 Mb). Disk editor, single / multiple disk copier / formatter / verifier.

DEA intelligent disassembler

This is a powerful intelligent disassembler. Commented assembly code (QDOS or SMS notation): Traps (QDOS, SMS, WMan, Ptr Gen...), System variables, Basic variables, Error keys. Automatic decoding even with very long files (usually no user intervention needed). Automatic Toolkit keyword extraction.

Open World

Convert GIF, IFF, TIF images into QL 4, 8 colours or monochrome screens. All screen resolutions supported (VGA, SVGA...). Sources of the C converter are included and can be used/altered/improved for any purpose provided that an acknowledgement to Open World is written somewhere.

MasterBasic

This is a useful development & debugging utility for the SuperBasic or SBasic programmer. This is a full function freeware version.

Zexcel

Spectrum 48k/128k emulator for the extended environment written by Davide Santachiara and Marco Ternelli. ZeXcel is the first and only ZX emulator for the QL which makes full use of the extended environment. It can emulate a Spectrum 128k +2 or a standard 48k ZX Spectrum with or without Interface 1 emulation.

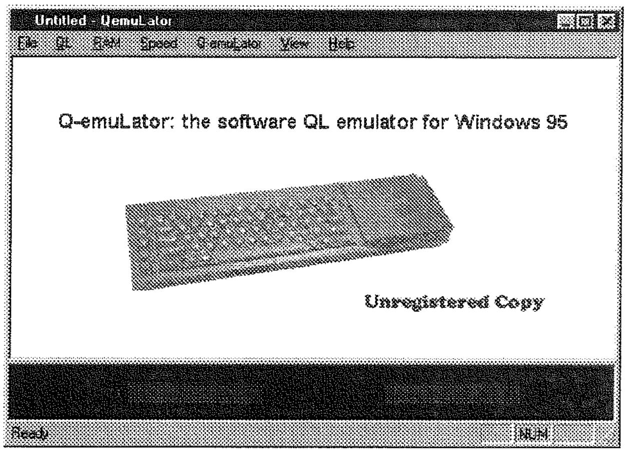
QL Emulators CD / Dilwyn Jones

The second Beta test release of the QL Emulators CD was produced as this was being written (mid November), with the final release expected some time during December. The CD-ROM is being produced in ISO-9660 format and will include emulators for Linux, Windows, DOS, Amiga, Atari ST and Apple Mac, all gathered into one convenient medium.

Linux - QLay and uQLx Windows - QemuLator (shareware), QPC2 demo and QLay DOS - QLay Amiga - QDOS4Amiga and QDOS Classic Amiga Atari ST - QeM Apple Mac - QemuLator

There will also be a large collection of QL freeware, public domain and shareware on the CD-ROM, along with copies of QL ROM images and the QL manual plus various other useful QL-related information and documentation along with copies of Quill, Archive, Abacus, Easel and Xchange. The CD-ROM is intended to be freely copied for all QLers, ex-QLers (to try to woo them back to the QL!) and even at those into "retro-computing" for whom QL emulators provides another platform.

QL Emulators CD will initially be available at low cost from Q-Celt Computing and I hope that in time those with CD-Writers will copy it for others too in the interest of increasing QDOS and SMSQE useage. By all means charge a small fee for copying, postage etc to cover your costs, but it should not be sold for profit - the only profit made will be from increased product sales if this CD attracts more people to the QL.



Q-emuLator: the software QL emulator for Windows 95

Unregistered Copy

Dilwyn Jones

The Beta Release 0.31 of Per Witte's ASM2HTM conversion program from assembler to html is now on my website's Other Software Page, plus a link for you to email him feedback as it is a Beta version.

Claus Graf

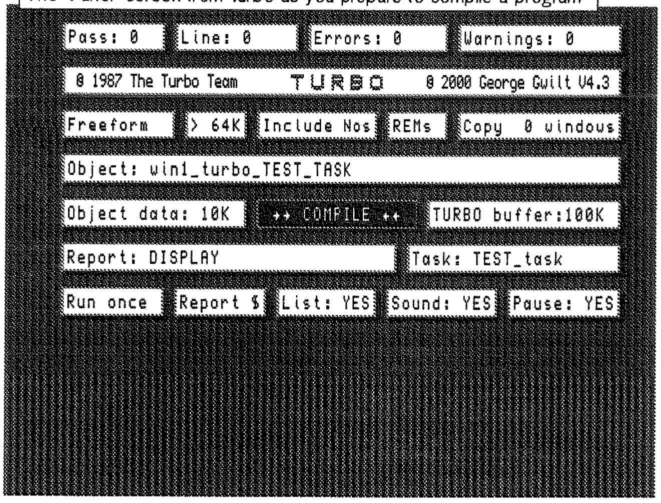
JPEG support in pqiv. I have uploaded a new version of the image viewer pqiv for Q40. It is now possible to decompress JPEG images. Both RGB and greyscale can be displayed. If you like to check it out, go to **www.q40.de**, please. The JPEG displaying is quite handy in connection with Simon N Goodwins Digicam software (that is still beta at the moment, but it works on a SMSQ/E/Q40 with 115200 baud connection speed, it gets developed for QDOS classic and therefore works also on that OS).

The New TURBO Compiler

Dilwyn Jones and George Gwilt

With the version 4.3 release of Turbo from George Gwilt, we now have a good BASIC compiler available free of charge for the QL.

Note: In this article I will use the expression S*Basic to refer to both SuperBASIC (QDOS) and SBASIC (SMSQ/E). SBASIC is a much enhanced version of QL SuperBASIC and comes as part of the SMSQ package on the QXL and the SMSQ/E system from Jochen Merz Software. SMSQ/E is the newer enhanced version of the QL operating system.



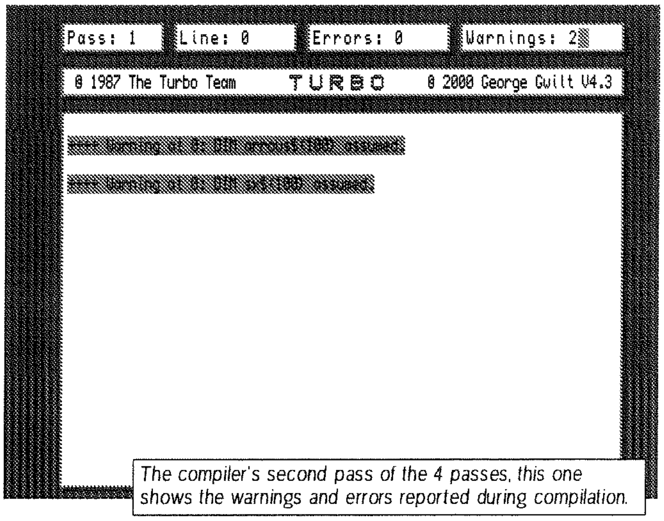
The 'Panel' screen from Turbo as you prepare to compile a program

```
Pass: 0   Line: 0   Errors: 0   Warnings: 0
© 1987 The Turbo Team   TURBO   © 2000 George Gwilt V4.3
Freeform   > 64K   Include Nos   REMs   Copy  0 windows
Object: win1_turbo_TEST_TASK
Object data: 10K   ++ COMPILE ++   TURBO buffer:100K
Report: DISPLAY   Task: TEST_task
Run once   Report $   List: YES   Sound: YES   Pause: YES
```

George Gwilt has been hard at work updating the Turbo Compiler recently, and a version 4.3 release at the time of writing was available from my website (goto **http://www.soft.net.uk/dj/software/other/other.html** and scroll down to the Turbo Compiler) and from some software libraries. The compiler is now freeware and has been extensively updated to be more compatible with SMSQ and SMSQ/E (no more need to patch the startup code in compiled tasks for example). For example, the compiler's windows no longer part disappear in mid-compilation when used to compile within pointer environment.

The compiler itself consists of just two tasks - PARSER_task which scans through your S*BASIC program (note: Job 0 only, may not work correctly with daughter SBASICs) and begins the compilation process, and CODEGEN_task which generates the actual compiled machine code task you will later execute.

In addition, there is also the Turbo Toolkit, a short piece of code which provides various extensions to BASIC. This comes in two versions, **TURBO_TK_CODE** for QDOS systems and **TURBO_SMS_CODE** for SMSQ/E system. Turbo Toolkit is now at version 3h27, thanks to the sterling work of David Gilham and Mark Knight in updating it. There are some new commands, and existing commands added in the Digital Precision release 3 have now been more fully documented and better supported by the compiler.

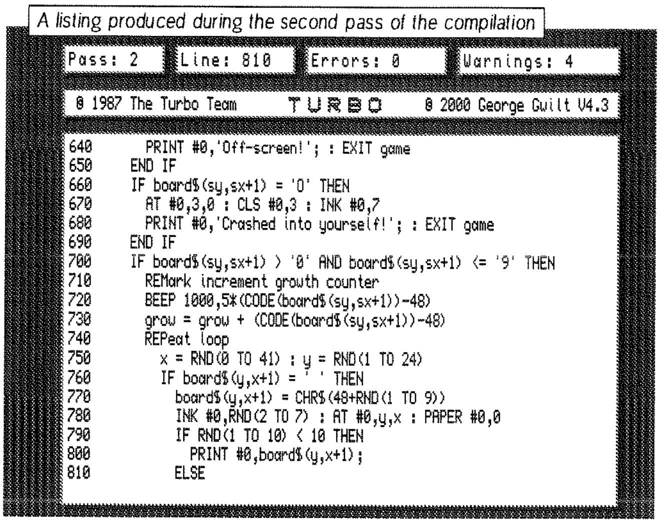


```
Pass: 1   Line: 0   Errors: 0   Warnings: 2
© 1987 The Turbo Team   TURBO   © 2000 George Gwilt V4.3

**** Warning at 0: DIM array$(100) assumed.
**** Warning at 0: DIM sx$(100) assumed.
```

The compiler's second pass of the 4 passes, this one shows the warnings and errors reported during compilation.

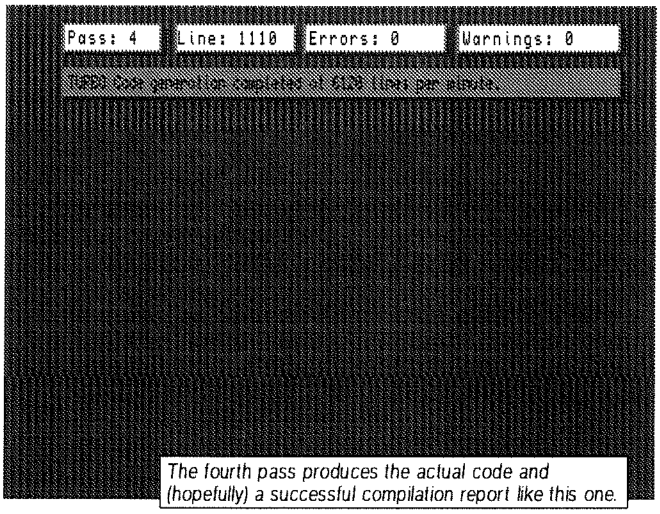
QLers who are already familiar with the earlier Digital Precision releases of Turbo will be well accustomed to some of the little details you have to pay attention to when compiling programs. For example, BASIC normally passes parameters to procedures by reference - that is, if the values of parameters are changed in that procedure, the variables passed to that procedure will also be updated unless they were part of an expression. By default, Turbo does not behave quite the same - variables parameters are treated as if passed by value and not by reference UNLESS you explicitly use the REFERENCE command to change this behaviour. This type of behaviour is still used by the current release of the compiler, so you will not need to change your way of slightly altering programs to work with Turbo. You still need to be careful to include 'name' parameters such as filenames in quotes to avoid possible problems when compiling such names. You still cannot put expressions in DATA statements in a compiled program and all RESTORE line_number lines MUST exist! Strings must be dimensioned as single dimension arrays - if you do not explicitly declare a maximum string length with the

DIM command, Turbo will assume that the maximum length of that string is 100 characters.


A listing produced during the second pass of the compilation

```
Pass: 2   Line: 810   Errors: 0   Warnings: 4

 © 1987 The Turbo Team     TURBO     © 2000 George Gwilt V4.3

640         PRINT #0,'Off-screen!'; : EXIT game
650       END IF
660       IF board$(sy,sx+1) = 'O' THEN
670         AT #0,3,0 : CLS #0,3 : INK #0,7
680         PRINT #0,'Crashed into yourself!'; : EXIT game
690       END IF
700       IF board$(sy,sx+1) > '0' AND board$(sy,sx+1) <= '9' THEN
710         REMark increment growth counter
720         BEEP 1000,5*(CODE(board$(sy,sx+1))-48)
730         grow = grow + (CODE(board$(sy,sx+1))-48)
740         REPeat loop
750           x = RND(0 TO 41) : y = RND(1 TO 24)
760           IF board$(y,x+1) = ' ' THEN
770             board$(y,x+1) = CHR$(48+RND(1 TO 9))
780             INK #0,RND(2 TO 7) : AT #0,y,x : PAPER #0,0
790             IF RND(1 TO 10) < 10 THEN
800               PRINT #0,board$(y,x+1);
810             ELSE
```

The Turbo Toolkit has added several facilities which can only be capitalised on with the new release of Turbo. For example, manifest constants are now possible - instead of saying LET colour = 2 when you want red ink, you can predefine names for those numbers to save having to look them up each time:
LET red% = 2
Manifest constants can also be used for strings.
The toolkit now implements LONGINTEGER and LONGINTEGER$ functions for handling 32 bit integers in the internal format. The previous toolkit included facilities for handling floating point numbers, integers and strings in internal format, but the facility to handle long word numbers (as used by POKE_L for example) was sadly missing. Just to make up the numbers, a new POKE_F procedure pokes a 6 byte floating point number into memory in the internal format at a specified address, which should be even.
Minerva integer tokenisation is also now supported, and the binary and hexadecimal notation used by SMSQ/E SBASIC can also be compiled. The previous limit of 16 channels per task has now been increased to 32. A new set of directives prefixed with TURBO_ allows you to set buffer size, task name etc (the compiler's front panel options) and can be defined from within the program so that you don't need to remember and set the panel options each time you compile a program. Array dimensioning restrictions in the original compiler have now largely been removed or relaxed. Numeric and string variables are now set to zero and null respectively.


The fourth pass produces the actual code and (hopefully) a successful compilation report like this one.

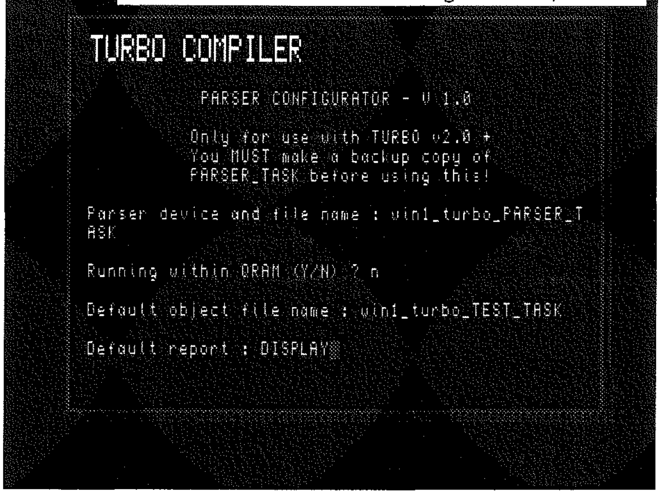
```
Pass: 4   Line: 1110   Errors: 0   Warnings: 0

TURBO Code generation completed at 6120 lines per minute.
```

FWINDOW% is a version of the WINDOW command implemented as a function. This means that you can try to open a large screen window and if that fails, resort to open a QL sized screen window instead, giving you a fairly simple method of handling windows on the hires displays now possible on modern QL systems.
A new feature is the DEBUG construct. This is an aid to adding debugging code to programs. You can specify 10 levels of DEBUG - once the DEBUG_LEVEL command sets a level higher than the numbers in the DEBUG statements (which can appear anywhere in programs) code can be conditionally included and excluded from a compiled program for testing purposes. It takes a bit of getting used to, but this is potentially a useful feature for programmers who need to test critical and complex code structures.
Turbo Toolkit, as with all versions since 3.00 from Digital Precision, is now fully ROM-able (i.e. contains no self-modifying code). The same cannot be said of Turbo compiled code, however. The toolkit can now be used with the QLiberator compiler if you wish, and is said to be so compatible with SMSQ/E that you can even install Turbo Toolkit as a module for SMSQ/E to remove the need to load it each time you start SMSQ/E. I am not sure how many people will use this facility, but it's there if you know how to add modules to SMSQ/E!
Some of the toolkit extensions have now been improved in terms of speed. For example,

SEARCH_MEMORY is now twice as fast under some conditions when used in interpreted programs, thanks to the use of index table code suggested by George Gwilt, one of a number of ways in which George Gwilt has contributed to the improvements in the compiler and toolkit alike.

When distributing programs compiled with Turbo, there used to be the restriction that you had to include a special 'runtime' version of the Turbo Toolkit, not the full version (it was usually found as a file called RUNTIME_EXTS as part of the program package). Now that the Turbo Toolkit is freeware, this restriction has been dropped and the full Turbo Toolkit version can be included free of charge with any compiled program which needs it. Although, of course, if your compiled task uses just standard S*BASIC commands and no special toolkit facilities, it is still possible to compile stand alone tasks as before. In fact, the word Task is in fact a bit of a misnomer, as the correct QDOS terminology is in fact a 'job' (in Tony Tebby-speak).

This release of the Turbo compiler is a great step forward for the QL. I personally started using Supercharge as my first compiler, then progressed to Turbo when that came out, then progressed to QLiberator. I'll probably stick to QLiberator as my compiler of choice largely because most of my existing programs were compiled with QLiberator and my whole programming environment is geared up for QLiberator at the moment. When I get around to updating some of my older Turbo compiled programs, I'll certainly still be using the new Turbo. As with most alternatives in life, you'll probably find that one compiler is great for some programming purposes, while the other is better suited for other tasks.


TURBO_CONFIG, which is used to configure the compiler itself.

```
TURBO COMPILER

              PARSER CONFIGURATOR - V 1.0

              Only for use with TURBO v2.0 +
              You MUST make a backup copy of
              PARSER_TASK before using this!

Parser device and file name : win1_turbo_PARSER_T
ASK

Running within QRAM (Y/N) ? n

Default object file name : win1_turbo_TEST_TASK

Default report : DISPLAY
```

Whichever you choose to use, Turbo and QLiberator are both extremely useful programming tools, which along with the free C68 compiler for C programmers give the QLer a good choice of tools for programming. If the SuperBASIC-C Port software ever gets updated, this will bring C68 and the BASIC compilers even closer together.

Using the Turbo Compiler

The slight fly in the ointment at the moment is that neither the toolkit nor the compiler has a full manual, although George Gwilt is looking into producing a manual for the compiler, while I'm looking at producing a short manual for new users of the Turbo Toolkit. Given the size of the original Digital Precision Turbo manual (presumably still copyright material, apart from perhaps being somewhat out of date?) both on paper and on disk in the special offer bundles offered by DP before they stopped advertising, producing such a full and complete manual will be no mean task.

The following is George Gwilt's brief introduction to compiling programs with Turbo, so that you can see what is involved if you have never used Turbo before. This text is included with the Turbo compiler as an appendix to the UPDATE_TXT file supplied. This is enough to get you going, but there's quite a few panel settings for example which requires some knowledge of the workings of the compiler, such as the greater than or less than 64K settings, the Freeform and Structured settings etc. Unless you are familiar with these settings, stick to the ettings George suggests which will give you the best chance of generating successful compiled tasks while working in the dark, even if it is not the best output possible all the time.

How to use Turbo

Turbo consisits of the two executable programs
PARSER_TASK
and
CODEGEN_TASK.
The instructions:

```
EXEC <directory>PARSER_TASK
EXEC <directory>CODEGEN_TASK
```

will compile the SuperBASIC program currently loaded. In the case of SMSQ it is the program in Master Basic which is compiled.

Alternatively, the command CHARGE, one of Turbo Toolkit's extensions, can be used. This has the same effect as

```
EXEC PARSER_TASK
EXEC CODEGEN_TASK
```

Therefore, for CHARGE to work, both PARSER_TASK and CODEGEN_TASK must be in the same directory and that directory must have been set by PROG_USE or Turbo Toolkit's equivalent, DEFAULT_DEVICE. For example, if the two programs are on win1_turbo_,

```
DEFAULT_DEVICE win1_turbo_
```

will allow CHARGE to operate correctly.

Note that the command CHARGE will not work if given in one of SMSQ's daughter basics.

What happens next

Provided that the Turbo Toolkit extensions have been loaded, PARSER_TASK presents a front panel. If you press SPACE immediately, or indeed at any other time that the middle box called COMPILE is selected, the compilation starts.

Otherwise you can alter the various settings by using the arrow keys to access the required box. To change the settings you may use the space bar (S) to toggle the options available, the up/down arrows to alter numbers (U) or the ENTER key to allow editing of an item (E).

Settings	*U/S/E*	*Comments*
Object data	U	Dataspace for the resulting program
TURBO buffer	U	Amount of ram used by
Turbo Object	E	Name of compiled program
Freeform/Structured	S	"Structured" means that the SuperBASIC program must not have any instructions following END_DEFine except DATA, REM, DEBUG, DEBUG_LEVEL REFERENCE (see below), DEFine FuNction and DEFine PROCedure. "Structured" is more efficient than "Freeform".
› 64K/‹ 64K	S	For large programs (like PARSER_TASK itself) › 64K is needed.
Include/Omit/Display Nos	S	Leave at "Include"
BRIEF/REMs/FAST	S	Leave at "BRIEF"

Copy n Windows	U	n = 0: no windows
		n = 1: copy #1
		n = 2: copy #0 and #1
		n = 3: copy #0, #1 and #2

Report	E	DISPLAY puts diagnostic output to the screen ‹filename› accepts the diagnostics
Task	E	The job name of the compiled program
Run once/Resident/Quit now?	S	Leave at "Run once" or "Quit now?" and ENTER to stop immediately
Create/Ignore/Report $	S	Leave at "Create $"
List	S	-
Sound	S	-
Pause	S	-

The comment "Leave at . ." effectively means "Read the full manual, if you can't do that it is safest to use the suggested option".

[We may consider putting Turbo on a future QL Today cover disk if there is sufficient interest. Let us know, please - Editor]

Me and SMSQ

Stephen Poole

I have been using SMSQ for about one year, and am very satified with it. Using it was like buying Turbo ten years ago, and dicovering an acceleration of program execution of about twelve times over. Since using SMSQ I seldom use Turbo, except where it offers useful functions not found elsewhere. I don't use the unnamed loop feature as this gets very confusing with nested loops. The PEEKing and POKEing keywords are useful, although not as good as Turbo's MOVE_MEMORY command. Quite often I need to peek into system tables, where the addresses are given in Jan Jones's Superbasic Handbook. It is a pity that no one has printed the equivalent lists for Sbasic.

The Sbasic Executable Thing is beyond me, as are Things in general. At the moment I am dabbling with multitasking, and am experimenting with the Event-handler. I hope the system is as neat as Turbo's job-handling.

The device driving facilities are clear, although I regret that there is no microdrive support, as I have 70 cartridges which I need to access frequently, so I cannot auto-boot Sbasic at the beginning of a session.

All in all there is very little to say about SMSQ, because the system is so smooth. As with the pointer environment, the latest version seems to incorporate all the best features to have appeared over the last few years. One wonders

what the forthcoming updates will be incorporating. A must for everyone.
Finally, here is a listing of Wipes_bas, a Cuedark module that I temporarily lost, and which is short enough for publication. When you run it, please dont think that your QL has crashed, as it will generate all manner of tweed patterns. Press a key to stop the program.

```
100 REMark SAVE as WIPES_bas (CueDark module), by S.Poole, v30oct2000.
110 REMark On SMSQ machines, just LRUN or EXEC flp1_WIPES_bas
120 REMark On non-smsq systems, replace QUIT by STOP.
130 :
140 OPEN#1,con_4: WINDOW 512,256,0,0: PAPER 0: BORDER 0: MODE 8: CLS
150 z=0: z%=z: u=1: h%=1: t%=2: e=128: a=255: a%=a: w%=512: L=256: l%=L
160 REPeat loop
170    c%=RND(a%): k%=c%: IF RND(1): k%=L-c%
180    g%=RND(a%): q%=g%: IF RND(1): q%=L-g%
190    aa=RND(1): bb=RND(1): cc=RND(1):  dd=RND(1)
200    IF (aa+bb+cc+dd)=0: GO TO 190
210    OVER -1: IF RND>.3: OVER 0
220    FOR s=4,2,1
230        IF RND>.5
240           b=L-s: st=s
250           FOR y=z TO b STEP st
260               y%=y: j%=a%-y%
270               IF aa: BLOCK w%,h%,z%,y%,k%
280               IF bb: BLOCK w%,h%,z%,j%,q%
290               i%=y%+y%: u%=l%-i%: IF u%<0: u%=-u%
300               IF cc: BLOCK t%,l%,i%,z%,c%
310               IF dd: BLOCK t%,l%,u%,z%,g%
320               i$=INKEY$(#1): IF i$<>'': EXIT loop
330           END FOR y
340         END IF
350     END FOR s
360     RANDOMISE DATE
370 END REPeat loop: MODE 4: QUIT
```

Gee Graphics! (on the QL?) - part 19

H. L. Schaaf

Matrices using Cramer's rule

What use can we make of matrices?

Solving simultaneous linear equations has been but one of the many uses of matrices. We will explore this use and it gives us an excuse to try the function and procedures from last month's listing.

There is one method using matrices for solving linear equations called Cramer's method, yet another method that makes use of the inverse of a matrix, and many other methods that use variations of Gaussian elimination.

Cramer's method was described in an appendix of his book "Introduction a l'analyse des Lignes Courbes algebriques", printed in 1750. This was more than a century before matrices were introduced by Sylvester and Cayley in the 1850's. What is now called "Cramer's method" had been described earlier in 1545 by Cardan. Cramer's book was taken up in the schools and widely taught and therefore Cramer's name is connected with the method. Gabriel Cramer was described as friendly, good-humoured, pleasant, an all-around nice person. He travelled throughout Europe and corresponded with the famous mathematicians of the time.

I've not found a reference as to when the "inverse approach" was developed, nor by whom. Can anyone tell us?

In the first decade of 1800 Gauss used an algorithm dating from at least 200 BC for successive elimination of unknowns and back substitution. His systematic description for solving six equations with six unknowns earned the name "Gaussian elimination" for the method. Many others have made variations that continued to polish and refine the method.

The Cramer's rule and inverse matrix are considered OK for small matrices, but Gaussian methods and their variations are favored for large matrices. It will be interesting to run some timing comparisons as to how long it takes for various methods on some QL setups.

Cramer's method makes use of determinants of matrices. First we get a determinant from a matrix of the unknown coefficients. Then we substitute the constants vector in place of the column for an unknown, one unknown vector at a time, find the determinant of that matrix and divide by the previously found coefficients determinant to get the answer for that unknown.
I'll try to set up a simple example.

Let's say we have three unknowns, x, y, and z. To make it easy, x = 2, y = 3, z = 5. Also we have three equations involving these unknowns. For example, these could be the equations for 3 planes in space.

We can set them up so a constant is on the right of the = sign:

```
4x + 7y - 2z =  19
3x - 5y + 4z =  11
1x + 4y + 3z =  29
```

Or the equations can be made homogeneous by expressing them as:

```
4x + 7y -2z - 19 = 0
3x - 5y +4z - 11 = 0
1x + 4y +3z - 29 = 0
```

which sets them up so that a zero is on the right of the = sign.

To find the values of x, y, and z we first set up the matrix, which we will call U for the coefficients of the unknowns:

```
( 4   7  -2 )
( 3  -5   4 )
( 1   4   3 )
```

and find the determinant which we can call Det U, or use the notation |U| and find that |U| = -193

then to find x we replace the column 4, 3, 1 with the constants vector C = (19, 11, 29) to form the matrix, which we will call X

```
( 19   7  -2 )
( 11  -5   4 )
( 29   4   3 )
```

and find determinant = |X| which is -386, x = -386/-193 = 2

then to find y we replace the column 7, -5, 8 with vector C to form the matrix, which we will call Y

```
( 4  19  -2 )
( 3  11   4 )
( 1  29   3 )
```

and find determinant = |Y| which is -579, y = -579/-193 = 3

then to find z we replace the column -2, 4, 6 with vector C to form the matrix, which we will call Z

```
( 4   7  19 )
( 3  -5  11 )
( 1   4  29 )
```

and find determinant = |Z| which is -965, z = -965/-193 = 5

These 3 planes meet at the point where (x, y, z) = (2, 3, 5).

Obviously a great little chore for a computer. Simply give it the data and let it do the work, and present the answers. Of course we could have a problem if the Determinant of U turns out to be zero, or if it becomes so small or so large that it taxes the accuracy and precision of our computer. As a check, we can plug our answers into the equations and see how closely the results match to the given canstants.

Load the matrix routines "MatFunPROCs_BAS" from the previous article and make them more general with these changes:

1 - Change Line 20050 to read:
   20050 LOCal n, i, swap_row

2 - Create a line # 20055 identical to line
    # 20490,
    20055 n = DIMN(mat_name,1)

Then merge the listing "Cramer_bas", and run it to see what answers you get.
Try making small (and large) changes to the DATA and notice the effect.

Also take a look at matrix routines in the QUANTA library. There is an assembly language version "MOPSIQL" by C. Gradwell on disk MA02, and a basic version "MATRICES_BAS" by Tom Bladon on disk MA01. You could probably revise "Cramer_bas" to work with these matrix programs from the QUANTA library.

Next time we will try the inverse matrix approach. We get nearly the same answers, but in less time.

Listing "Cramer_bas"

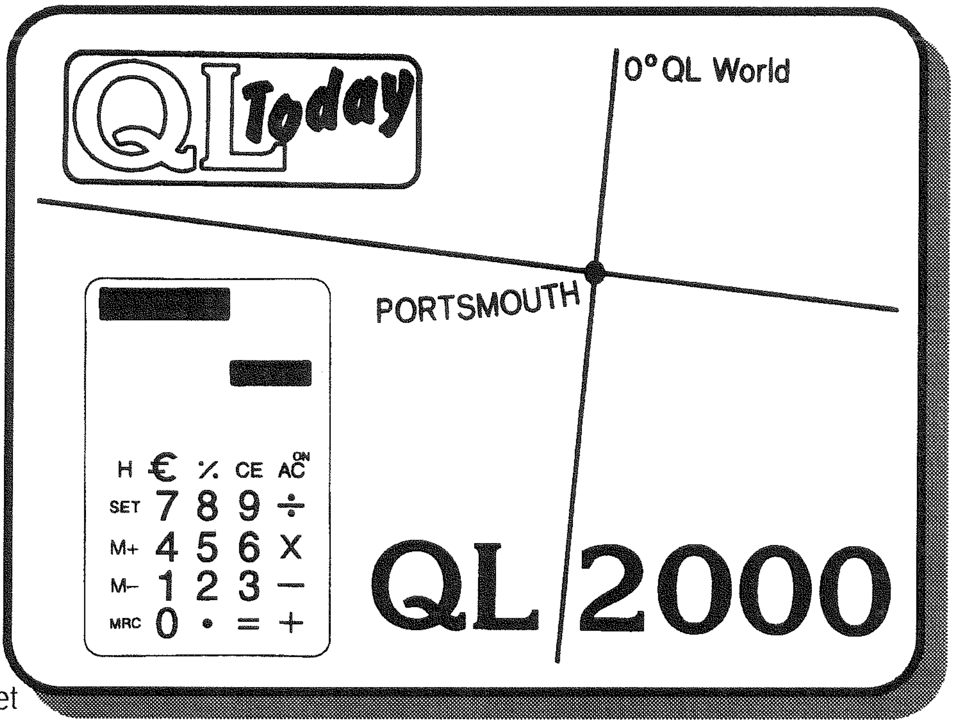
```
100 REMark Cramer_bas
110 REMark for GG19 HL Schaaf 3 Oct 2000
120 REMark using QL to solve linear equations with matrices
130 REMark MATFuNPROCS_bas needs to be merged with this
140 :
150 REMark sample set:
160 REMark 4x + 7y - 2z =  19
170 REMark 3x - 5y + 4z =  11
180 REMark 1x + 4y + 3z =  29
190 REMark answers are x = 2, y = 3, z = 5
200 :
210 WTV : PAPER 0: INK 4: CLS
220 unknown$= "xyz"
230 matrices$ ="XYZ"
240 RESTORE 260
250 :
260 DATA 3
270 DATA 4,  7, -2,   19
280 DATA 3, -5,  4,   11
290 DATA 1,  4,  3,   29
300 :
310 REMark could also set up to allow for input from user
320 REMark number of variables == number of equations
330 REMark by entering and checking entries before "solving"
340 :
350 READ n
360 REMark need this as n in MAT_DET, should have had a local n
370 REMark see MAT_INV as example of how locals were defined
380 REMark otherwise could refer to it as 'rank' for a square matrix
390 :
400 REMark store the coefficients of the unknowns in matrix U
410 DIM U(n,n)
420 REMark store the constants in C, a vector
430 DIM C(n)
440 REMark provide for the answers in A, a vector
450 DIM A(n)
460 FOR i = 1 TO n
470   FOR j = 1 TO n
480     READ U(i,j)
490   END FOR j
500   READ C(i)
510 END FOR i
520 :
530 PRINT, "Unknown coefficients matrix U"
540 MAT_SHOW U
550 A(0) = MAT_DET(U)
560 PRINT, "Determinant of matrix U = |U| = ";A(0)
570 IF NOT(A(0)) : PRINT "sorry, no solution ! ":STOP
580 :
590 REMark working_array W
600 DIM W(n,n)
610 FOR i = 1 TO n
620   MAT_COPY U, W
630   FOR j = 1 TO n
640     W(j,i) = C(j)
650   END FOR j
660   PRINT\, "matrix ";matrices$(i)
670   MAT_SHOW W
680   A(i) = MAT_DET(W)
690   PRINT, "Determinant of matrix ";matrices$(i);
700   PRINT  " = |";matrices$(i);"| = ";A(i),
710   A(i) = A(i)/A(0)
720   PRINT unknown$(i);" = |";matrices$(i);"| / |U| = ";A(i)
730   PAUSE 100
740 END FOR i
750 PRINT \A(1 TO n),\\
760 :
770 REMark now cross check results ?
```

```
780 PRINT "Checking out results to confirm answers"
790 PRINT "eq#","sum","given","difference"
800 FOR i = 1 TO n
810   sum = 0
820   FOR j = 1 TO n
830     sum = sum + (U(i,j)*A(j))
840   END FOR j
850   PRINT\i, sum, C(i), sum - C(i)
860 END FOR i
870 :
880 REMark end of listing Cramer_bas
```

Missed Opportunity, here is a Second Chance

Jochen Merz

If you did not make it to the QL2000 show: the present for all QL-Today subscribers at the show was a nice mousepad, slightly smaller than DIN-A4, with a QL 2000 souvenir print and - the highlight - with built-in calculator and currency converter. Both display and solar area are built neatly into the mousepad, the "keys" are also part of the mousepad-surface. A nice present, which I immediately ordered when I saw it.



I ordered significantly more than the number of people who said they would come and pick it up. In the end we were left with zero, because of the high number of people who decided to attend but did not tell us.

But, visitors asked me if it would be possible to buy a few more. The problem: I need to order at least 40. Roy and Tony would take a few, but the question is: is there enough interest to get 40 together?

I will collect orders until end of December. If we get enough together, I'll order another batch - otherwise we have to forget about it, which would be a shame. Price depends slightly on the amount I order - it will definitely be under £6 each. The mousepad is not just a "QL Collectors Item", but extremely useful too.

This is how it looks (reduced in size), the pad is white. Please note the "border" of the calculator is not a real border - the whole surface is one area, except for the display and solar area.

Missing Postcards!

We have mailed all QLers in our database recently and invited them for the recent QL shows. We also asked everybody who is still interested to return the postcard to us, so that we can compile a list of interested people - who will be invited to shows etc. We were hoping to get a better feedback, especially hoping to get email addresses so that we can save a bit of money on postage! Surprisingly, we got many returns from non-QL-Today readers, but not so many from QL-Today readers. If possible, return the card with your email address, please!

Q40 - an intermediary assessment

Peter Graf

The by now not so new successor to the QL, the Q40, has already become a mature total system and proves to be a motivating force for the development of hard- and software. This may be the right time to make an intermediary assessment. Let's start by looking back: The first public appearance of the Q40 was nearly three years ago. That was during the QL meeting at Eindhoven during the autumn 1997. I came with my one and only prototype. In a strenuous fir of work I had got the Hardware working and I had written quite some software, practically a mini operating system together with some demo software, to be able to test -and show- all features. I was able to use the harddisk and also to show some high resolution colour images. Claus had even written a driver to show QL fount characters on the screen. Tony Tebby attended that meeting and had many a discussion. I had never spoken to him personally, and didn't know whether the new hardware would really be of any interest to him. Once he passed by our table and friendly hit the monitor with one hand, but without saying anything. In the afternoon, Tony Tebby came to have a look at the hardware and to get information on many details. I didn't know it then, but by then Tony Tebby had already decided by himself that he would develop an SMSQ/E for the Q40. Quite some time was to pass before he would really start on it, but that day was an important milestone on the road to a new QL.

The current QL hardware at that time probably was an Aurora card to replace the QL motherboard, SuperGoldCard, Qplane backplane, Qubide IDE interface together with some chips from the original QL, a keyboard interface etc. This was useable, but you had to tinker quite a bit, to build such a system. Some of the parts weren't even produced any more.

For me, many parts of my wish list remained unfulfilled. For example, to be able to display many colours in high resolution, and a fast graphics system. A faster CPU. More memory. Faster peripherals. Real sound. An expansion bus for which one would be able to buy sensible cards. I wanted to have a compact - and stable- total QL system, a real new computer built all in one.

So, the Q40 hardware now existed. It had the best possibilities to offer all of this. Now the time to develop the Q40 operating system began. The pioneer in this field wasn't Tony Tebby, but Mark Swift. He was first able to adapt an operating system to the Q40. That was the QDOS Classic, which supports many of the features of the Q40, amongst others IDE harddisks and Sound. It is highly compatible with the QL, especially for older software. QDOS Classic can be gotten for free, including source code, for example in the internet at **http://pages.unisonfree.net/ mswift/files/QZ.**

In the meantime, I had fully developed the Q40 hardware, and had already produced a first series, to provide developers with motherboards. A year later, it finally was time: During the autumn of 1998, again at the Eindhoven show, Tony Tebby showed the first version of SMSQ/E for the Q40. That was really a great moment, especially as Tony Tebby intended to use the Q40 as his own main development platform. Now even the doubters and procrastinators could no longer ignore that the Q40 was a reality. Qbranch decided to produce the Q40 in series. There were some more delays, but for a year and a half everybody can buy a Q40.

I'm really happy that the Q40 was able to play a role as catalyser for the development of some important hard and software in the QL field, most importantly, of course, the new QL high colour drivers. At last, there existed a new, 68040 compatible system with sufficient graphics capabilities to fulfil this wish of many a QL user, in time for the start of the new millennium. I have to thank Tony Tebby again for this.

The Q40 was the first system to run the new colour drivers, and with high speed. You should just imagine that the Q40, at a resolution of 1024x512 with 65536 colours has to transfer 30 times more graphics information than in the good old mode 4. So new graphics programs saw the day, to display images in different file formats, to edit colourful sprites, to convert into other formats or produce background images. Even producing 3D graphics with the ray tracing program Povray is now so much faster, because the Q40 contains a floating point coprocessor (FPU). A luxury which is missing on the SuperGoldCard or the QXL.

In other fields, too, there were quite some developments. For example, Tony Tebby intro-

Q Branch

PROGRAMMING

QD 98	£ 45.00
QD + QBasic	£ 59.00
QD + Qliberator + QBasic	£ 100.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 30.00
Easyptr pt 1 & 2 (together)	£ 30.00
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 15.00
QMon / JMon	£ 22.00
Basic Linker	£ 19.00
DISA 3	£ 31.00
QMenu	£ 14.00

Text 87

£ 79.00

Typset 94	£ 29.00
Fountext 94	£ 39.00
2488 drivers	£ 29.00
Epson ESC/P2 drivers	£ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handliing the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

So the festive season is on us once more and for those of you seeking gifts for your QL friends why not consider a copy of QPC 2 or other software now at Super Low prices? How about a trial subscription to QL Today for those members of your user group who are still wavering ? There are so many things to chose from.

We do have a few superHermes keyboard adaptors available at very low prices.

Full superHermes only £ 65.00 with keyboard adaptor.
superHermes Lite only £ 30.00 with keyboard adaptor.

Special seasonal offers :

Cueshell 2 £ 15.00

Easyptr parts 1 & 2 with free pointer tutorial disk £ 20.00

These offers only last until the last day of 2000. Buy now while stocks last

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00
(Includes Pointer and non-pointer driven versions)
(P.E. versions need Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)
Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC - New program !!

Convert text files from QL to PC formats and much more !

Only £ 10.00 Now with HTML support !

Spelling Crib : PD program £ 1.50 +SAE or Free if you buy all three programs

UTILITIES

FiFi 2	£ 18.00
QSup	£ 28.00
QSpread v2.04	£ 48.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 28.00

We are currently out of stock of the SuperBasic Reference Manual
Place your order now to get one as soon as it is reprinted

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.
compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !
£ 40.00
+ postage

Tel +44 (0) 1273-386030 fax +44 (0) 1273-381577
Mobile +44 (0) 7836-745501
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

Q Branch

Feeling out on a limb ?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

Please Note our new address and phone numbers

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

The ProWesS word processor

¶ Demo version £ 1.50 + postage
Full Registered version £ 18.00
Version 2.03 available now !

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

Few only

Super Gold Cards	**£ 125.00**
QXL	**£ Call**
Recycled superHermes	£ 80.00 *
Recycled Gold Card	£ 60.00 *
Recycled Aurora	£ 75.00 *
Aurora	£ 90.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.
14" and 15" monitors for the Aurora - Call.

SMSQ/E

Gold Card / Atari / QXL Version
£ 69.00
Various Atari versions : call for details
The Colour Drivers are running on the Q Branch Q 40 !
Coming to a computer near you in 2000!

QPC 2 colour beta version!
£ 82.00 (£ 67.00 SMSQ/E Owners)
Upgrade to the colour beta version now
Upgrades from QPC 1 & old version £ 27.00
(return master disk)
Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E

Q Branch Programs

The Knight Safe 3 *NEW VERSION!* £ 35.00
upgrades from previous versions £ 5.00

Q - Route v1.08C £ 25.00
Route finding programme *NEW VERSION!*

Q - Count £ 25.00
Pointer driven home accounting

The Fractal Collection !

This is a brand new program which will produce stunning animated fractal patterns. It will run on anything from a Gold Card to the Q40 and will be capable of using the power of the colour drivers when they are released. Complete with many example files and routines to design your own screens.

Only £ 35.00

Q Branch wish you all a Merry Christmas And a Happy New Year

VISA

We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.

MasterCard

duced for the first time the QL Sampling Sound System for the Q40, which enable the reproduction of sampled music or speech through the operating system. By now, several programs were developed for the Q40, which can work with and reproduce, different sound formats. This is possible because the Q40 has a simple but effective stereo sound hardware, with connections for earphones, active speakers or a hi-fi system. The Sampling Sound System has also been implemented for QDOS Classic.

There is also quite some progress for the remaining system software, such as a partitioning program, with which you can partition an IDE harddisk and create logical drives, directly under SMSQ/E. Tony Tebby has already mentioned in the documentation, but not yet implemented, some new concepts, such as the PROT_MEM command, with which you can protect some parts of the memory against unauthorised access. This uses the Q40 memory management unit (MMU), which again is a feature that didn't exist for earlier QL hardware.

It unfortunately is true that Motorola no longer develops the processors of the 68xxx series. Hence, all new QL compatible hardware developments can't compare on price with the PCs based on Intel processors, also not the Q40. Unfortunately, Motorola does not build an 800 MHz 68040 processor. On the other hand, if you see how much trouble working on an Intel computer, especially one running Windows, can produce, then a QL compatible hardware is still very justified. A Q40 boots QODS or SMSQ/E in a few seconds, without subjecting the harddisk to a long dance - just put it on and off you go. And the processing speed is so high that, until now, I have only had very positive comments - despite the fact that in current versions, SMSQ/E doesn't even use the Q40 cache in full. In earlier version, it was fully activated, so that Tony Tebby will hopefully be able to repair this again. You can also use QDOS Classic if, for example for calculations, you wish to make use of the entire speed of the Q40. With QDOS Classic, the user is able, through some Basic commands, to configure the cache a bit more precisely.

The biggest lack of the QL scene, that of the lack of software, unfortunately, can't be repaired by a hardware, even though the Q40 did give some impulses. However, here the Q40 has another possibility where no relevant QL software is available, and this through the Q40 Linux. This operating system was ported to the Q40 by the programming genius Richard Zidlicky, and it is being constantly updated. It is freely available, for details see, e.g. in the Internet under

http://www.geocities.com/Silicon Valley/Bay/ 2602/q40.html

There is a proper Q40-Linux distribution, available on a CD-ROM, even including a graphical installation programme. You may start Q40-Linux directly from QDOS Classic or SMSQ/E. The Q40 is thus the first and only QL-style computer for which Linux is also available, and in a very professional and stable version. An enormous mass of software runs under it, from Internet access to XWindows and Web browsers such as Netscape up to professional graphics programs such as Gimp. The Q40, under Linux, can play MP3 music files, read Audio CDs digitally and much more. Of course, for a pure QL user without knowledge of Linux, this is less interesting, but it does show that many things are possible with this computer - and that you don't necessarily need an Intel PC to use Linux..

Incidentally, the Q40 can also drive CDROM drives and writers, and with the assistance of a plug-in card you can also use Ethernet! Unfortunately, there are no QDOS or SMSQ/E drivers for these devices, but under Linux, everything already works OK.

So, this is an assessment of how the Q40 was developed, and what it can already do. To look into the future, I confirm that, in a foreseeable time, there will be an even faster platform for QDOS and SMSQ/E, with the Q60. The Q60 will have a Motorola 68060 CPU at 60 MHz or maybe a bit more, but apart from that, it will have features that are very close to those of the Q40. Thus, if you don't need a maximum speed, you will be well served with the Q40, and don't need to wait until the Q60 is in production.

Finally, a tip for those who want to be closer informed through the Internet: The central website for the Q40 is

http://www.q40.de

Thanks to Jonathan Dent's new TCP/IP software we will hopefully soon be able to access this also under QDOS or SMSQ/E ...

Programming Prowess in SBasic - Part 4

Wolfgang Lenerz

We are going to try to write another example program today. This is actually very closely inspired by the example 1 program contained in Prowess. It sets up a small window containing two loose menu items and two "info objects". You can hit or do the items. Hitting or doing the second item has absolutely no result, whereas hitting or doing the first item changes the text in the two info objects, to show how often the item was hit or done:

```
set_windows
test1
:
DEFine PROCedure test1
LOCal object, hit%,hits,dos,times$,mhit$,mdo$
LOCal loop%,add_info
:
: REMark first initialise some variables
  mhit$="You have hit the item ":times$=" times":hits=0
  mem=0:object=0:hit%=0
  mdo$="You have done the item ":dos=0
:
: REMark now make some strings, note the CHR$(0) at the end!
:
  my_hit$=mhit$& hits&times$&CHR$(0)
  my_do$=mdo$&dos&times$&CHR$(0)
:
REMark now create the outline object
:
  outl=PWcreate(0,PW('TYPE_OUTLINE'),
              PW('OUTLINE_QUIT'),
              PW('OUTLINE_SLEEP'))
:
REMark now create the item objects
:
  item1=PWcreate(outl,PW('TYPE_LOOSE_ITEM'),
                  PW('LOOSE_TEXT_COPY'),"Hit or Do me",
                  PW('LOOSE_ACTION_DO'),DO_ROUTINE,
                  PW('LOOSE_ACTION_HIT'),HIT_ROUTINE)

  item2=PWcreate(outl,PW('TYPE_LOOSE_ITEM'),
                  PW('LOOSE_TEXT_COPY'),"Hitting or Doing me
                                         will do nothing",
                  PW('LOOSE_ACTION_DO'),DO_ROUTINE,
                  PW('LOOSE_ACTION_HIT'),HIT_ROUTINE)
:
REMark now we create two infostring objects
:
  info1=PWcreate(outl,PW('TYPE_INFOSTRING'),
                  PW('INFOSTRING_TEXT'),my_hit$,
                  PW('INFOSTRING_AUTOSIZE'),0
  info2=PWcreate(outl,PW('TYPE_INFOSTRING'),
                  PW('INFOSTRING_TEXT'),my_do$)
:
REMark the main loop
:
  REPeat loop%
    mem=PWactivate(outl,mem,object,add_info,hit%)
    IF NOT mem:EXIT loop%
    SELect ON object
      =item1
```

```
       SELect ON hit%
         =0:hits=hits+1
            my_hit$=mhit$&hits&times$&CHR$(0)
            PWchange info1,PW('INFOSTRING_TEXT'),my_hit$
         =1:dos=dos+1
            my_do$=mdo$&dos&times$&CHR$(0)
            PWchange info2,PW('INFOSTRING_TEXT'),my_do$
        END SELect
    END SELect
  END REPeat loop%
:
  PWremove outl
END DEFine test
```

The program starts out, easily enough, with the set_windows procedure, which I have already commented on in an earlier instalment of this series. It then calls the test1 procedure.

In that procedure, after having initialised some variables, we create the outline of the window. The window is supposed to have a quit item, and a sleep item.

(Please note that, for the sake of clarity, I have split up the parameters for the function over several lines. Unless you have the Basic Linker, you cannot do that, and should always have the parameters for the function on the same line).

The creation of the outline object is straightforward enough:

```
     outl=PWcreate(0,PW('TYPE_OUTLINE'),
           PW('OUTLINE_QUIT'),
           PW('OUTLINE_SLEEP'))
```

In other words, we tell Prowess that we want to create an object, that its owner is 0, that we want to create an object of the OUTLINE type, and then we pass this objet an 'OUTLINE_QUIT' tag and an 'OUTLINE_SLEEP' tag. This then returns the object we created in the 'outl' variable. Actually, to be more precise, this returns the 'Object ID' of the object created in the 'outl' variable. The outline object is created.

This is perhaps a good point to have a closer look at the outline type itself.

The Outline type

As was already mentioned, an outline is intended to be the first object when a window is constructed, all other objects then being owned by that outline. Unless you specify otherwise, an outline consists only of a title (which by default displays the program name), with below that a separator line. If you create nothing but an outline, that is what you will see.

Optionally however, there can be some loose menu items at the right or left sides of the title. These are: a Quit, Sleep, Help, Do, Wake and/or an Info item. The info item can be defined by the programmer (both text and action). The other items have their normal standard use (quit quts, sleep put the program to sleep behind a button etc...).

An Outline also always contains two empty (and invisible) boxes at the left and right. These boxes are, of course, also Prowess objects, The Object ID of these objects can be obtained, and other objects can be created inside these boxes (by specifying that one of these boxes is the owner of the object to be created). Thus the functionality of the outline can also be extended.

It now only remains to be seen what kind of tags can be used with the outline type. You will, of course, find all of the tags that can be used with the type in the manual, but I would like to give some explanations here.

Generally speaking, and this is true for all of the Prwess types, tags are used in three kinds of operations: (i) when creating an object with PWcreate, (ii) when changing an object with PWchange and (iii) when querying an object with PWquery.

In most cases, change tags can also be used when creating the object and vice-versa (there are some rare exceptions). Query tags are only used for queries.

The change (and creation) tags for the Outline type

The following are the tags that can be used when creating or changing an outline:

PW('OUTLINE_SLEEP')

This tag indicates that the outline is to contain a sleep item which, when actioned by the user, puts the program to sleep as a button. This tag needs no parameters. By default, the sleeping program will display its name, but that can be changed (see next tag). You should only use this tag in the "primary window" of your program. The primary window is the first window open for your program, i.e. in general the first outline created and activated. Also, you should not have any screen channel open when this is called, other than those opened by Prowess.

PW('OUTLINE_SLEEP_TEXT')
This tag also ensures that the outline contains a sleep item. Contarry to the previous tag, this tag needs one parameter, a string, which is the text to be is displayed by the button when the program is put to sleep. As for the previous tag, you should only use this tag in the primary window of your application.

PW('OUTLINE_QUIT')
This ensures that the outline has a quit item. When pressed, the program quits. It is possible to have an automatic confirmation request before quitting. By default, the action of the quit item depends on the quit confirm status (see below).

PW('OUTLINE_ACTION_QUIT')
This tag will attach a user defined action to the quit item. The quit item must already exist before passing this tag (i.e. pass the previous tag first). This tag needs one parameter, which is the pointer to the user defined action routine, preferrably QUIT_ROUTINE (this will be explained later). Please note that, in this case, the object returned from the PWactivate call will be the QUIT object, not the outline object. You can query the outline to obtain the quit object with one of the queries.

PW('OUTLINE_QUIT_CONFIRM')
Set the quit confirm request status. This tag needs one parameter, which is either 1 (=TRUE, a confirmation request is made) or 0 (=FALSE = no confirmation request is made). The default quit action uses this status to determine whether a confirmation request should be popped up before quitting. If this status is set to 0 (i.e. FALSE), then the window will be exited as soon as the user has actioned the Quit item, otherwise, a window will pop up to query whether the user is really sure s/he wants to quit the window.

PW('OUTLINE_QUIT_KEYPRESS')
Attach a keypress to the quit item. The quit item must already exist before passing this tag (i.e. pass the PW('OUTLIEN_QUIT') tag first). The parameter should be a number corresponding to the key to be used (e.g. 27 for ESC, or CODE('x') if pressing 'x' shoud quit the window). By default the quit item has no keypress attached to it, which is pretty unnerving, since the user then has to use the moue to action that item. Normally, you would use ESC (= 27) as keystroke for the quit item.

PW('OUTLINE_INFO_TEXT')
This tag indicates that an info item is to be included in the outline, and it sets the text for this info item. There is thus one parameter to go sith this tag. The parameter, a string, is the text which will be displayed in the item. There is only one info item per outline (but, of course, you can create a loose item with the text set to "Info" if you want).

PW('OUTLINE_ACTION_INFO')
This tag also indicates that an info item object is to be included in the outline and it sets the action routine for this object. If no info item existed already when this tag is passed, then the info item is created by this tag, and the text in it will be "info", else this will apply to the info item already created with PW('OUTLINE_INFO_TEXT')). The parameter is an action routine. It should be INFO_ROUTINE (mmore of this later), and is the action routine for the info item.

PW('OUTLINE_ACTION_DO')
This tag ensures that the outline contains a do item (object), which can be activated also by a DO keypress (ENTER or right mouse button). The tag is followed by a parameter, which should be DO_ROUTINE, and is the action routine for the Do item.

PW('OUTLINE_ACTION_WAKE')
This tags ensures that the outline contains a wake item, which can be activated also by a the ussual Wake keypress (CTRL F2) keypress. The tag is followed by a parameter, which should be WAKE_ROUTINE.

PW('OUTLINE_TITLE_TEXT')
Set the title for the outline. The parameter following this tag is a string. By default, i.e. if this tag is not used, the title will be the program name.

PW('OUTLINE_HELP')
Make sure a help item is included in the window. The default action for the help item is to execute the Prowess reader (which should be loaded as a resident extension - to make it into an executable thing). The file which has to be displayed, the directory where it can be found and the position in the file can be specified by the PW('OUTLINE_HELP_xxx') tags (see below). This tag requires no parameters.

PW('OUTLINE_ACTION_HELP')
Assign your own action routine to the help item in the outline. If there was no help item yet, then it will be created. The tag needs a parameter which should be HELP_ROUTINE (more of which later).

PW('OUTLINE_HELP_FILE')

Specify which help file should be loaded when the help item is indicated. This will automatically reset the position in the file (so the file will be displayed from the start). This tags needs a parameter, which is a string with the filename.

PW('OUTLINE_HELP_POSITION')

Specify the position in the current help file which should be displayed when the help item is indicated by the user. This tags needs a parameter which is a string (i.e. the Prowess reader will go to that string).

PW('OUTLINE_HELP_DIRECTORY')

Specify the directory where the help file should be searched. The parameter is a string.

The query tags for the Outline type

Quiery tags are used when querying the object with PWquery. The query tags for the Outline type mainly allow you to get at some implicit objects, i.e. some objects which are created at the same time as the outline:

PW('OUTLINE_BOX_LEFT')

To allow the user to modify the behaviour and look of the outline object, there is always an empty (and invisible) box at the left corner in the outline. This empty box is an object itself, and you can may put other objects in it. The box object is returned by this query (i.e. you say

`box_left=PWquery (outline, PW('OUTLINE_BOX_LEFT'))`

and this returns the object, or Object ID, in box_left).

PW('OUTLINE_BOX_RIGHT')

Likewise, there is also always an empty box at the right in the outline. The object is returned by this query.

There are also some other objects, which you can create explicitly when creating (or changing) the outline, such as a Quit item etc... Here again, these objects are not returned by the creation of the Outline or any change operation for the outline, so you can query for the object IDs, so that later you can SELect on these objects:

PW('OUTLINE_OBJECT_QUIT')

This returns the quit object in the outline.

PW('OUTLINE_OBJECT_INFO')

This returns the info object in the outline.

PW('OUTLINE_OBJECT_DO')

This returns the DO object in the outline.

PW('OUTLINE_OBJECT_WAKE')

This returns the wake object in the outline.

PW('OUTLINE_OBJECT_HELP')

This returns the help object in the outline.

Back to our example program

Once the outline is set up, we create several other objects. Note how they are all owned by the outline object.

First, we create two loose items with some text in them. Loose items are, of course, a Prowess Type, just like an outline, and we will cover them in more detail in the next instalment of this series. Notice how the tag used to put the text in the items (PW('LOOSE_TEXT_COPY') is a "_COPY" tag, so that we can give direct strings as parameters. For ***item1*** we also indicate routines to call when the item is "hit" (***HIT_ROUTINE***) or "done" (***DO_ROUTINE***). These don't exist for the other item, which is why hitting or doing it will have no effect. These routines will be explained in the next instalment of this series.

After this, we set up two infostring objects. An "infostring" is another Prowess type. These two objects will contain the text which says how many times you have hit or done ***item1***. For the first infostring, I have set AUTOSIZE to 0 (i.e. False - by default it is True, i.e. 1). We will come back to this later, but here is a short explanation: When AUTOSIZE is set for an object, then changing this object in any way (e.g. changing the text is displays) causes the entire window to be redrawn.

As you can see, this makes a difference: when the text in the first item is changed (after a hit), the window is not redrawn, since AUTOSIZE is not set. If the text in the second item is changed, the window is redrawn. This is because the length of the text might have changed and it might have become much longer. This means that the text might no longer fit in the object. If it doesn't, then, IF AUTOSIZE IS TRUE, the object containing the string is made larger, and then, of course the window must also be made larger - and redrawn. This is why the window is also redrawn if AUTOSIZE is TRUE...

The main loop

These few lines are enough to set up the window! Once this is done, we can actually have it do something, like drawing it on the screen. So, we come to the main loop of the program. All Prowess programs will have such a loop (or something similar). We first call the ***PWactivate*** function which draws the window on the screen and waits for your actions. At the first call, mem is 0, which is as it should be. Later on, it is automatically maintained by the system.

Of course, mem is also returned by the ***PWactivate*** call. The ***PWactivate*** call comes back whenever you actioned QUIT, or the first item. It won't come back for the second item, since we haven't given this item a hit or do routine.

When the ***PWactivate*** call comes back, we first check whether mem is 0 or not. If it is, we should quit the loop since by definition, mem will only be 0 if the user indicated that he wants to leave the window. In that case, we leave the loop, remove the outline object with ***PWremove*** (which also removes all the other objects owned by the outline object i.e. all other objects in this case), and the procedure is finished.

If mem is not 0, we should check which object was hit. Here we know that this object was necessarily item1, so actually we don't really need to check this. On the other hand, we check whether the object item1 was hit or done. The parameter hit% to the ***PWactivate*** function will reflect that: a ***DO_ROUTINE*** will set this to 1, whereas a ***HIT_ROUTINE*** will set it to 0. So, according to whether item1 was hit or done, we change the text in info1 or info2. The text is changed with the ***PWchange*** command, and the info object is automatically redrawn. For the object info2 this also automatically redraws the entire window, since AUTOSIZE was left on.

There are a certain number of concepts which I have left unexplained for the moment, notably the xxx_ROUTINE commands. We shall discover these in the next instalment of this series, when we have a closer look at looes items.

The QL Religion Collection

A Review by Al Boehm

I like to verify ideas and facts. So after some Mormons visited me, I looked in The Book of Mormon to see for myself. You don't have The Book of Mormon? Well neither do I as a book, but I do have it on The QL Religion Collection.

What is it?

This collection contains a vast amount of information about religion. Further more it contains a tremendous variety of ideas from topics ranging from atheism, to the numerous ways to say the rosary. It has seven different translations of the Bible:
American Standard Version
King James (Quill and text)
Koinee Greek New Testament
Deuterocanonical Books
The MicroBible
NIV Bible
New King James Bible

The bible books are stored as separate files in a directory so that there would be no trouble loading them into a standard QL although some require 256k ram. This size limitation is not a big problem on the QXL or emuLators. Based on the work of Howard Clase, the Koinee Greek New Testament directory has Quill patched to show Greek letters and a patch to print in Greek. The King James directory has besides the text, graphics of the pre-flood patriarchs.
In addition, The Collection has a wealth of information to help you study the Bible:
The Authorship Of The Bible
Bible Companion program
Bible Dictionary
Biblical Cross References
Critique of Translations
King James Bible Influence
Myths About The Bible King James Bible history

But it doesn't stop there. A large variety of Christian documents are included. The list below is titles of directories. Many contain five, ten, or more documents related to the topic:
Praying The Rosary
Catholic Church History
Chesterton: Orthodoxy, etc.
Christian Jokes!
Jesus Is Coming Soon
Book of Common Prayer (US)
Joe Crews sermons
Does God Love You?
MegaSermon pack
Assortment of religious texts
Who Nailed What To The Cross?
Parakletos: Holy Spirit
The Conversion Of Paul
Women in the Christian church
Church-style Dates Converted
The Dead Sea Scrolls
Sermons On The Card (playing)
The Problem Of Suffering
The Tower Of Babel
Sermons by Reuben A. Torrey

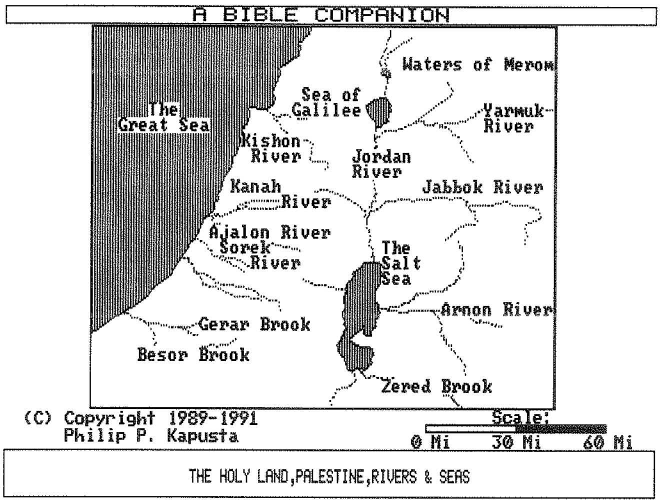
Each of these is a directory! They can contain many different items. For example, the Assortment of religious texts contains 17 different articles ranging from "Clothing as Legalism" to "Spiritual Warfare". Each directory also contains information on where it came ›from and other background information. Some are historic texts and might be very difficult to find in a large public library. For example, "Sermons On The Card" about the evil or not of playing cards was written in 1529.
Most items are freeware or public domain but there are a few shareware items. For example, there is a Hymnal Database ($20 shareware) which runs on Archive. It contains title, author, composer, subject, church season, scripture, style, format, meter, theme, subject matter and the first line of the first and second verse for the five Hymnals: 1982 Episcopal Hymnal, Hymns For the Family of God, Lutheran Book Of Worship, Presbyterian Hymnbook, United Methodist Book Of Hymns, and United Methodist Hymnal. There is provision for adding items from additional Hymnals.
Also included is a wealth of information on other religions and Christian Sects:
King Asoka Edicts (Buddhist)
Atheism-related text files
The Baha'i faith
Is There A God? (Buddhist)
Buddha: Life&Teachings, etc.
The Teachings Of Phra Ajaan
The Eightfold Path (Buddhist)
Religions Of Ancient China
Confucius: The Great Learning
God The Known and Unknown
The Song Of Hiawatha
Bhagavad-Gita; + (Hindu)
Invention Of A New Religion
Various Islamic Text Files
Judaism text files
Kabbalah chariot prophecy
The Holy Koran
Kevin Solway writings
Marcus A. Antoninus stoic
The Book Of Mormon Mormon
Biblical Evidence
The Story Of The Mormons
Pagan Yule Customs, Ceremony
The Profits Of Religion
Christian Reincarnation
Religions Of The World
Sikhism
Ch'an Tao Chia: Essays

Want to know what the Stoics really believed? It's in Marcus A. Antoninus. Want to know what a cynic thinks of religion as a shield to privilege? It's in Upton Sinclair's "The Profits of Religion". How about "Christian Reincarnation" or how a new religion was started in Japan? - "Invention Of A New Religion"
Aside from text information, there is also included a very large number of different types of art work. For example, QL clipart screens in compressed and uncompressed format, Bible illustrations, Line Design clipart. My son has used some of this clip art in flyers announcing church events and to liven up some of the hymns I have written. See **http://ziplink.net/~boehm/music.html**

Programs

A number of viewer programs are included:

MiniView -simple viewer
Viewer 1.15 - more capability
MView 2.21 - pointer driven
Bitmaps - graphics viewer
Graphics Viewer-uses pointer



Also included are the utility programs for graphics compression, FileFinder to find a text string in a file (very useful for searching the Bible for a half remembered verse!), Phil Borman's SUB directory device driver, and Zip/Unzip. Plenty of good stuff here!

Origin and Use

Most of the material was extracted from various PC sources such as PD libraries and the Internet. They were then converted to QL format by Dilwyn Jones and George Morris. I can appreciate this was not a five minute task but a steadfast effort over several years.

The bulk of the material is in standard text files. You can use the supplied text readers or your own favorite text reader such as Arced or QD. There are a few files in Quill format.

The material in each directory was initially put on a floppy or set of floppies. Thus, programs are set to run from flp1 or in some cases flp1 and flp2. On my system, the CD-ROM is win5. (Yes, I have three large QXL hard drive directories plus a Syquest directory. It's hard to believe that 10 years ago I could carry in one hand the floppies that contained everything I thought was worth having.) Thus I used the recommended DEV substitution:

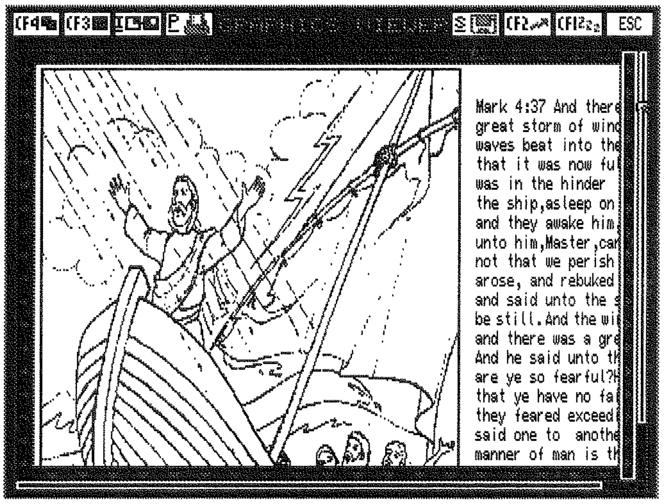
```
DEV_USE 1, WIN5_hymnal_
DEV_USE FLP
```

to run the hymnal archive program and access its data base. These commands switch any call to FLP1_ to DEV1_ but in addition DEV1_ is switched to win5_hymnal_. My CD-ROM has a sleep algorithm such that when it hasn't been called for a period of time, it stops. It then takes a few seconds to rev up when it is called. To get around this, I WCOPYed the hymnal directory to ram1. Then used:

```
DEV_USE 1, RAM1_
DEV_USE FLP
```

The simpler RAM_USE FLP did not work on the QXL or on Q-emuLator?

The CD-ROM worked fine on the QXL and on my PC laptop with Q-emuLator 2.1. DIR win5_ worked and individual files read OK. I tried using QD to see if the QMenu system would be able to find the directories and files. It worked fine on the QXL but QD on the Q-emuLator had trouble picking up the directories. If the directory was typed in, it worked OK. No problem, I just ran two copies of Q-emuLator with the directories listed on one and QD running on the other. I could have just multi-tasked, but this way the Q-emuLators were side by side.



Mark 4:37 And there
great storm of wind
waves beat into the
that it was now fu
was in the hinder
the ship,asleep on
and they awake him
unto him,Master,car
not that we perish
arose, and rebuked
and said unto the s
be still.And the wi
and there was a gre
And he said unto th
are ye so fearful?H
that ye have no fa
they feared exceed
said one to anothe
manner of man is th

Availability

I was a little late with this review. Every time I started to check out a feature, I became engrossed with the topic and had to read all about it! We all owe George Morris, Dilwyn Jones and Darren Branagh (for producing it) a hearty thank you for bringing yet another capability to the QL. George also told me that Simon N. Goodwin provided him some invaluable technical assistance on the fly at their sub-group meetings.

Now I am prepared. If someone shows up at my door from the Sikhism religion, I can check out any claims that are made. Plus I occasionally play music in various churches. Think how nice it will be to know what is in their hymnal on the subject of, say, repentance. If I have to give a talk at a church activity, I can start off with a nice joke from the Christian Jokes directory. Also I heard there was a squabble at a nearby Catholic Church on the right way to say the rosary. Now I can show them several dozen ways!

The QL Religion Collection on CD-Rom can be obtained from Q-Celt Computing. The cost is

10 pounds plus postage. If you do not have a CD-ROM capability but have a Syquest drive in either QXL or Qubide format, I will make you a copy of as much as will fit at the cost of the disk and postage plus 10 pounds. My address is 2501 Ermine Dr., Huntsville, AL 35810 and my email: **boehm@ziplink.net**

Bug fix first

Per Witte has pointed out a potential problem in my article about the Maths Stack in QDOS. I mentioned that A1 is not set to a usable maths stack pointer until such time as parameters have been fetched. Per has pointed out that if zero parameters were fetched, then A1 ids still no good and will crash the system if used.

I have tested this and found it to be correct. Per allows his users to default all of the parameters, while I never have. If I detect that no parameters were passed, I bale out with a BAD PARAMETER error, so I have not had the reported crash.

If you wish to allow your users to supply no parameters, and wish to use the maths stack to return a result, then always remember to check the count of parameters passed in D3.W and if zero, set A1 to the value in BV_RIP(A6), like this :

```
zero_params
       cmpi.w  #0,d3             ; Did we get ZERO parameters
       bne.s   some_params       ; We got some
       move.l  bv_rip(a6),a1     ; We need to initialise the
                                 ; maths stack
       bra.s   skip_params       ; Skip over the parameter
                                 ; handling stuff
some_params                      ; Process parameters on
                                 ; maths stack here
```

Another gotch to watch out for. Thanks Per.
Norman.

LTdis - part 4

Norman Dunbar

In the previous instalment of the QLTdis program, I left you typing away merrily - there was quite a lot of typing to get done - this time we just have to wade through a set of algorithms which we will use to decode the individual instruction types. Not much (if any) typing in this instalment.

As with much computer programming, certain assumptions are made about what other bits of the code will have done prior to getting where we are right now - this is no exception. I am assuming that the main disassembling loop will have carried out some work and set values in certain registers prior to calling these type decoding sub-routines. I shall list these assumptions so that we all know where we stand.

On entry to the following routines, it is assumed that:

1. The address for this instruction has been printed followed by a space (or two) so address 1024 will therefore appear as 'L00000400 ' - all addresses are in Hex and all are prefixed by the letter 'L'. When we decode a branch or jump instruction, the 'label' we use as the destination will be in this format too.

2. The instruction's string has been copied from the table to the output buffer. So if we are decoding the instruction 'ADD.W #1,D0' then the output buffer at A5 holds the text 'ADD' and D6.W (the length of the output string) currently holds the value 3.

3. D7.W holds the op code as does D0.W. We will be working with the value in D0.W but may well need to 'repair' it at times when we mask out bits or whatever else we have to do to get from op-code to fully decoded instruction. We don't simply reload it from memory as the pointer to the address where we got it from originally has been moved on as we decode data words etc. D7.W will be our safe storage.

4. A6.L holds the address of the NEXT byte after the op code. This is so that we can calculate destination addresses for BRA instructions etc. A6 will be modified as we decode each word of an op-code and all its data.

5. PC_ADDR(A4) holds the address of the opcode itself. This is the address of the first word of the current instruction that is being decoded.

6. A5 points at the first free space in the buffer where we are building up our decoded instruction. As in assumption 2 above, A5 holds the address of the character after the final 'D' in 'ADD' ready for the next character to be stored there.

7. D5.W is set to zero. It WILL eventually hold the actual size of the instruction we are decoding so our assumption 2 instruction the size is WORD and D5 will eventually hold the value 2. The values stored in D5.W will be zero (for undefined), 1 (for byte), 2 (for word) and 4 for long. The various type decoding routines need to set D5 before calling the effective address routine as certain effective addresses need to know the instruction's size in order to correctly read the next few bytes of data from the memory area being disassembled.

8. D6.W holds the number of characters stored in the output buffer so far. Using our assumption at 2 above, D6.W will be holding the value 3 as there are 3 characters in 'ADD'.

9. When finished decoding, each type sub-routine jumps back to the same place in the code where the hex bytes of the instruction will be printed.

10. Each line of output will have the following 'fields':

```
Laddress   hex-codes   decoded instruction
ascii version of hex-codes
```

The address will be 8 characters plus 1 for the 'L'. There can be up to 10 bytes in an instruction so we will need enough space for 20 characters. The decoded instruction varies in size - assume 20 characters (but I may adjust this later!) Another 10 bytes maximum for the ascii codes.

This is 59 characters just for the data and allowing 2 bytes between 'fields' we now have the following:

```
00 - 08 = address
09 - 10 = spaces
11 - 30 = hex bytes
31 - 32 = spaces
33 - 52 = decoded instruction
53 - 62 = ascii bytes
```

Assumptions are now listed so on we go with the decoding.

When I first documented the various types, I presumed that they would follow in order of complexity with type 0 being the simplest up to type 30 being most complex. Now that we are working through each one decoding as we go, It appears that some simple ones are more difficult to decode than at first glimpse. Not to worry - we can do it!

In each type, I give one or two examples of what we are decoding. This is not exhaustive and is there only to remind you what is being decoded. Hopefully the following pseudo code can be easily read and easily converted into assembly languages when we come to do the next batch of typing. Pseudo code, for those who may not have come across it yet, is simply a mixture of English and code which describes what is to be done in a more readable form than pure code or pure English. (apologies to non-English speakers at this point.)

The instruction families - decoded

Type 0

This is the simplest instruction type. By the time we get to this point, all the work has been done and we are ready to print the instructions hex codes etc.

```
Example 'RESET'.
Done.
```

Type 1

This type has a single word of data following the op-code but only the lowest byte is required. All we have to do is extract this byte and add it to the instruction (in hex) then finish off the instruction with ',CCR'. See 'QLTdis part two' for details of each instruction type and the instructions in its 'family'.

```
Example 'ANDI #$data,CCR'
Get the word at (A6)+ into D4.
Convert D4.B into hex and add it to the op-code buffer.
Add ',CCR' to the op-code buffer.
Done.
```

Type 2

This type also has a word of data and this time both bytes are used. All type 2 operations are acting upon the Status Register.

```
Example 'ANDI #$data,SR'
Add the word at (A6)+ to the op-code buffer in hex.
Add the string ',SR' to the op-code buffer.
Done.
```

Type 3

Type three family instructions require a register number to be added to the string that we already have in the op-code buffer. As numerous instructions require this, we have extracted the code to a sub-routine which will be explained later. This routine adds the appropriate register details to the op-code buffer so no further work is required here.

```
Example 'SWAP Dn'
Call source register subroutine.
Done.
```

Type 4

Type four is the TRAP #n instruction where 'n' can be any value between 0 and 15. We can use the hex routine if 'n' is between 0 and 9 but if it is greater than 9 we cannot or we get the number in hex when we must have it in decimal. All we do is test to see if 'n' is greater than 9 and if so, put a one in the buffer, subtract 10 from 'n' and then we can use the hex routine safely. The joy of reusing existing code!

```
Example 'TRAP #data'
Mask out bits 4 to 15 of D0.W as we only want bits 0 to 3.
D0.W is now a value from 0 to 15.
If D0.W > 9
   Add '1' (digit one) to the op-code buffer
   subtract 10 from D0.W - D0.W is now 0 to 5.
end if
Convert D0.W to hex and add it to the buffer (it will be 0
to 9 now !)
Done.
```

Type 5

Type five is the LINK instruction. This is quite easily decoded as follows.

```
Example 'LINK An,#$data'
Call source register subroutine.
Add ',#$' to the op-code buffer.
Add the WORD at (A6)+ to the op-code buffer in hex.
Done.
```

Type 6

The decrement and branch instructions are covered next. These have a condition code in bits 8 to 11 and are decoded in the following way.

```
Example 'DBcc Dn,label'
Mask out all bits except 8 to 11 - the condition code.
Shift D0.W right by 8 bits so bit 8 becomes bit 0.
Call the condition code routine passing the parameter in
D0.W (Preserves D0)
Add ' D' to the buffer. (Space D)
Call the source register routine.
Add a ',' to the op-code buffer.
Copy A6.L to A3.L
Add the word at (A6)+ to A3 (Remember the sign extends)
This is the address of the label to branch to.
Add an 'L' to the op-code buffer.
Convert A3.L to hex and add it to the buffer. We will
decode this as 'DBcc Dn,Lxxxxxx'
Done.
```

Type 7

Another simple instruction. BSR is decoded thus:

```
Example 'BSR label' or 'BSR.S label'
Copy A6.L to A3.L
If D0.B <> 0
   Add '.S ' to op-code buffer
   Add D0.B to A3.L (the sign will extend).
   This is the address of the label to branch to.
else
   Add ' ' to op-code buffer
   Add the word at (A6)+ to A3 (sign extended).
   This is the address of the label to branch to.
end if
Convert A3.L to hex and add it to the buffer. We will
decode this as 'BSR Lxxxxxx' or 'BSR.S Lxxxxxx'
Done.
```

Type 8

Another instruction with condition codes. As above with the DBcc instructions, we process it as follows noting the special case where the condition code is 0 which means BRA rather than BF which doesn't make any sense.

```
Example 'Bcc label' or 'BRA.S label'
Mask out all bits except 8 to 11 - the condition code.
Shift D0.W right by 8 bits so bit 8 becomes bit 0.
If D0.B = 0
   Add 'RA' to the op-code buffer (BRA instead of BF)
else
   Call the routine to convert D0.B into a condition code.
endif
Copy D7.W to D0.W.
Copy A6.L to A3.L
If D0.B <> 0
   Add '.S ' to op-code buffer
   Add D0.B to A3.L (sign extended).
   This is the address of the label to branch to.
else
   Add ' ' to op-code buffer
   Add the word at (A6)+ to A3 (sign extended).
   This is the address of the label to branch to.
end if
Convert A3.L to hex and add it to the buffer. We will
decode this as 'Bcc Lxxxxxx' or 'Bcc.S Lxxxxxx'
Done.
```

Type 9

Type nine is quite simple as the following description shows. Note that when we mask out D0.W leaving only the size bits in bits 6, 7 and 8 we don't bother to shift them we just test if D0.W is zero and if so the size is word otherwise it has to be long. Byte sized EXT instructions are not valid and we cannot use our 'size' sub-routine because this family has a non-standard value in the size bits.

```
Example 'EXT.size Dn'
Mask out all but bits 6 to 8 - the size part.
If D0.W = 0
   Add '.W' to the op-code buffer
else
   Add '.L' to the op-code buffer
end if
Add ' D' to the op-code buffer. (Space D)
Copy D7.W to D0.W.
Call the source register routine.
Done.
```

Type 10

Type ten is the MOVEQ instruction and is so simple to decode. The data is held in bits 0 to 7 of D0.

```
Example 'MOVEQ #$data,Dn'
Convert D0.B to hex and add to the op-code buffer.
Add ',D' to the op-code buffer.
Call the dest register routine.
Done.
```

Type 11

Type eleven is the Binary Coded Decimal instructions plus ADDX and SUBX. In the description in part 2, I mentioned that it was quite a tricky instruction to decode. Wrong ! By testing bit 3 of D7 we know whether the '-(Ax),-(Ay)' version or the 'Dx,Dy' version of the instruction is being used.

```
Example 'ABCD -(Ax),-(Ay)' or 'SUBX Dx,Dy'
If bit 3 of D7 is set
   add '-(A' to the op-code buffer
else
   add 'D' to the op-code buffer
end if
Call the source register routine. Preserves D0.
If bit 3 of D7 is set
   add ')' to the op-code buffer
   add ',-(A' to the op-code buffer
else
   add ',D' to the op-code buffer
end if
Call the dest register routine.
```

```
If bit 3 of D7 is set
   add ')' to the op-code buffer
end if
Done.
```

Type 12

Another simply instruction to decode once we know that bit 3 defines the direction of the MOVE to or from the USP.

```
Example 'MOVE An,USP' or 'MOVE USP,An'
If bit 3 of D7 is set
   add 'USP,' to the op-code buffer
end if
Add 'A' to the op-code buffer
Call the source register routine.
If bit 3 of D7 is clear
   add ',USP' to the op-code buffer
end if
Done.
```

Type 13

Yet another simple instruction.

```
Example 'CMPM.size (Ax)+,(Ay)+'
Call size subroutine to add size to op-code buffer.
(Preserves D0 and adds a space)
Add '(A' to op-code buffer.
Call the source register routine. Preserves D0.
Add ')+,(' to the op-code buffer.
Add 'A' to the op-code buffer.
Call the dest register routine.
Add ')+' to the op-code buffer.
Done.
```

Type 14

Back to more complex instructions for a while. The EXG instruction can be between two address registers, or two data registers or one of each. The value held in bits 3 to 7 of the op-code define the correct mode.

```
Example 'EXG Dx,Dy' or 'EXG Ax,Ay' or 'EXG Dx,Ay'
Mask out all but bits 3 to 7 of D0.W.
Shift D0.W right 3 bits.
Save D0.W on the stack - we need it again later.
If D0.W = 9
   add 'A' to the op-code buffer
else
   add 'D' to the op-code buffer
end if
Copy D7.W to D0.W.
If  the word on the stack (our old D0) = 17
   call dest register routine (preserves D0)
else
   call source register routine (preserves D0)
end if
Add ',' to op-code buffer.
If the word on the stack (our old D0) = 8
   add 'D' to the op-code buffer
else
   add 'A' to the op-code buffer
end if
If the word on the stack (our old D0) = 17
   call source register routine (preserves D0)
else
   call dest register routine (preserves D0)
end if
Tidy the stack by removing the old D0 word.
Done.
```

Type 15

MOVEP is the only type fifteen instruction. It is actually quite simple to decode.

```
Example 'MOVEP.size $data(An),Dn' or 'MOVEP.size Dn,$data(An)'
If bit 6 of D7 is set
   add '.L ' to the op-code buffer (dot L space)
else
   add '.W ' to the op-code buffer (dot W space)
end if
If bit 7 of D7 is set
   add 'D' to op-code buffer
   call dest register routine (preserves D0)
   add ',' to op-code buffer
end if
Add '$' to the op-code buffer.
Convert the word at (A6)+ to hex and add to the op-code buffer.
Add '(A' to the op-code buffer.
Call the source register routine (preserves D0).
Add ')' to the op-code buffer.
If bit 7 of D7 is clear
   add ',D' to op-code buffer (comma D)
   call dest register routine (preserves D0)
end if
Done.
```

Type 16

Back to a dificult one again. The shifts and rotates are the type sixteen family and there is much wailing and gnashing of teeth required to decode this little lot.

```
Example 'ASL.size Dx,Dy' or 'ASL.size ‹ea›'
Mask out all but bits 3 and 4 of D0. These bits define the correct instruction.
Shift D0.W right by 2 bits. D0 = 0, 2, 4, or 6
Point A3 at SH_TABLE.

The values in D0 are 0,2,4 or 6 and represent 'AS', 'LS',
'ROX' and 'RO'. Rather than have a table with size words in
it as well, we simply have a table of 2 character strings
and process D0 to see if an 'X' should be added
afterwards.
This makes life a lot easier and I am all in favour of that!
See SH_TABLE below.

Get the word at A3 + D0 into D4. D4 = 'AS' or 'LS' or 'RO' or 'RO'
Add D4.W to the op-code buffer.
If D0 = 4
   Add 'X' to the op-code buffer (We are doing ROXR or ROXL)
end if

Now we have the correct instruction, we need the
direction part of it.
If bit 8 of D7 is set
   add 'L' to op-code buffer
```

```
else
   add 'R' to op-code buffer
end if
```

D0 is now corrupted so we restore it from D7 and extract the size part.

```
Copy D7.W to D0.W again.
Call the 'size into D0' routine which sets D0 to the size bits.
```

Now that D0 has the size bits we can check to see which sort of shift/rotate we are doing. If D0 is three then we must be doing a memory shift/rotate which is always word sized. Otherwise, we are shifting/rotating a register.

```
If D0.W = 3
   This is a word sized shift in memory
   Add ' ' to the op-code buffer
   Move 2 (word) to D5 for size. Required by the effective
  address routine.
   Copy D7.W to D0.W again
   Call effective address routine (preserves D0)
else
   We must be shifting/rotating a register
   Copy D7.W to D0.W again
   Call size decoding routine (preserves D0 & adds a
space to the op-code buffer)
   if bit 5 of D7 is clear
      This must be a 'count in data' shift/rotate
      add '#$' to the op-code buffer
      mask out all but bits 9 to 11 of D0.W
      if D0.W = 0
         add '8' to the op-code buffer
      else
         call the dest register routine
      end if
   else
      This is a 'count in register' shift/rotate
      add 'D' to the op-code buffer
      call the dest register routine
   end if
   add ',D' to the op-code buffer
   call the source register routine
end if
Done.
```

```
SH_TABLE    DC.W    'ASLSRORO'
```

Type 17

All other types require the effective address decoding to be carried out. This is described below. The type seventeen family contains a large number of different instructions (see part two for full details) and in order that we know which ones, we have to carry out a bit more masking and checking - similar to how we got here in the first place from the main disassembly loop. First we test for the MOVE SR, ‹ea› and MOVE CCR, ‹ea› instructions.

```
Example 'MOVE SR, ‹ea›' or 'NBCD ‹ea›'
And D0.W with $FFC0
If D0.W = $40C0  (MOVE SR, ‹ea›)
   do nothing !
else
   if D0.W = $42C0 (MOVE CCR, ‹ea›)
      do nothing
   else
      add ' ' to op-code buffer
   end if
end if
Copy D7.W to D0.W
```

Regardless of the instruction being decoded, we are now at a position where we must decode the effective address. Before calling the effective address routine we need to set D5 to the size required. Only 'MOVE ‹ea› CCR' and 'MOVE ‹ea› ,SR' are allowed to use immediate data so these require the size to be 2 (word) even though the size specifier is not used in the decoded instruction. By setting D5 to 2 we can then call the effective address routine.

```
Copy D7.W back into D0.W
Set D5.W to 2 (word sized)
Call effective address routine
```

D7 is our storage for the original op-code word. We now AND D7.W with $FFC0 to discover which of the remaining instructions need some extra decoding. This is the MOVE ‹ea› ,SR and MOVE ‹ea› ,CCR instructions.

```
And D7.W with $FFC0
if D7.W = $44C0
   Add ',CCR' to op-code buffer
else
   if D7.W = $46C0
      Add ',SR' to op-code buffer
   end if
end if
Done.
```

Type 18

This family of instructions have one bit used to determine whether any words of data are involved. This is bit 14 which, if clear, means that we have to accomodate the data, otherwise we don't have any data involved.

```
Example 'ORI.size #$data, ‹ea›' or 'CLR.size ‹ea›'
Call the size decoding routine. (Preserves D0, sets D5 &
adds a space)
If bit 14 of D0.W is clear (Not CLR or NEG etc)
   call sub_mode 4 routine (#$immediate data) (preserves
   D0)
   add ',' to the op-code buffer
endif
Call the effective address routine. (D5 is set in the size
decode routine)
Done.
```

Type 19

ADDQ and SUBQ are next. These have the data value encoded into bits 9, 10 and 11 of the op-code word and allow values between 1 and 8 to be added or subtracted from an effective address.

```
Example 'ADDQ.size #$data, ‹ea›' or 'SUBQ.size #$data, ‹ea›'
Call the size decoding routine. (Preserves D0, sets D5 &
adds a space to the buffer)
Add '#$' to the op-code buffer.
Mask out all but bits 9 to 11 of D0.
If D0.W = 0
   add '8' to the op-code buffer
else
   Shift D0.W right 9 bits.
   convert D0.B to hex and add to the op-code buffer
end if
Add ',' to the op-code buffer.
Copy D7.W to D0.W again.
Call the effective address routine. (D5 = set from size
decode routine)
Done.
```

Type 20

The next two families cover the Bit manipulation instructions. As the actual op-code is defined in bits 6 and 7 we have extracted this decoding to a separate sub-routine which is called by both families. When we enter these two sub-routines the op-code buffer holds a 'B' only.

```
Example 'BTST #$data, <ea>' or 'BCLR #$data, <ea>'
Call the bit-op routine. (Trashes D0, adds CLR, TST, SET or
CHG to buffer)
Add '#$' to the op-code buffer.
Get the word at (A6)+ into D4.
Convert the BYTE in D4 to hex and add to the op-code
buffer.
Add ',' to the op-code buffer.
Copy D7.W to D0.W.
Set D5.W to 1 (meaningless value)
Call effective address routine.
Done.
```

Type 21

```
Example 'BTST Dn, <ea>' or 'BCLR Dn, <ea>'
Call the bit-op routine. (Trashes D0, adds CLR, TST, SET or
CHG to buffer)
Add 'D' to the op-code buffer.
Copy D7.W to D0.W.
Call the dest register routine. (Preserves D0).
Add ',' to the op-code buffer
Set D5.W to 1 (meaningless value).
Call the effective address routine.
Done.
```

Type 22

A relatively simple piece of decoding next for CHK, DIVS, DIVU, MULS and MULU.

```
Example 'CHK <ea> ,Dn' or 'DIVS <ea> ,Dn'
Set D5 to 2 (word size)
Call effective address routine (preserves D0).
Add ',D' to op-code buffer. (comma D)
Call dest register routine.
Done.
```

Type 23

Another easy one, LEA is the only member of this family.

```
Example 'LEA <ea> ,An'
Set D5 to 4 (long size)
Call effective address routine.
Add ',A' to op-code buffer. (comma A)
Call dest register routine.
Done.
```

Type 24

Back to the complicated stuff again. Type 24 covers the ADDs and SUBs not already catered for by other families. 'ADD' and 'ADDA' share the same value after masking, as does 'SUB' and 'SUBA' which means a bit of twiddling to extract the correct instruction.

Example 'ADD <ea> ,Dn' or 'ADD Dn, <ea>' or 'SUBA <ea> ,An'

First we set D0 to the size bits and if we detect a size of 3 then we are dealing with ADDA or SUBA and we add an 'A' to the buffer before decoding the real size which is held in bit 8. Prior to calling the effective address routine, we need to set D5 to 2 or 4 for word or long as appropriate.

```
Call the 'size into D0' routine. (Trashes D0)
If D0.W = 3 (ADDA or SUBA only)
   add 'A' to the op-code buffer
   if bit 8 of D7 is clear
      add '.W' to the op-code buffer
      set D5.W = 2 (word)
   else
      add '.L' to the op-code buffer
      set D5.W = 4 (long)
   end if
   add ' ' to the op-code buffer
   copy D7.W to D0.W
   call the effective address routine (preserves D0)
   add ',A' to the op-code buffer
   call the dest register routine
   done.
```

The size was not 3 so we are dealing with an ordinary 'ADD' or 'SUB' etc. First we manipulate the value in D0 to set it correctly for the effective address routine prior to copying it into D5 which is where the effective address routine expects it to be.

```
else (ADD <ea> ,Dn or ADD Dn, <ea> etc)
   D0.W = 0 1 or 2 for byte, word or long
   Shift D0.W left by 1 place (0 2 or 4)
   if D0.W = 0
      set it to 1 (byte)
   end if
   Copy D0.W to D5.W (correct size specifier)
```

Now we restore D0 from D7 and start decoding the remaining instructions in this family.

```
   Copy D7.W to D0.W again.
   Call the size decoding routine (preserves D0)
   if bit 8 of D7.W is set (ADD Dn, <ea> or SUB Dn, <ea>)
      add 'D' to the op-code buffer
      call the dest register routine
      add ',' to the op-code buffer
   end if
   Call the effective address routine (preserves D0)
   if bit 8 of D7.W is clear (ADD <ea> ,Dn or SUB <ea> ,Dn)
      add ',D' to the op-code buffer (comma D)
      call the dest register routine
   end if
   Done
end if
```

Type 25

Much of this code is similar to type 24 so no further description is required.

```
Example 'CMP.size <ea> ,Dn' or 'EOR.size Dn, <ea>' or
'CMPA.size <ea> ,An'

Call 'size into D0' routine. (trashes D0)
If D0.W = 3 (CMPA.size <ea> ,An)
   add 'CMPA' to the op-code buffer
   if bit 8 of D7 is clear
      add '.W' to the op-code buffer
      set D5.W = 2 (word)
   else
      add '.L' to the op-code buffer
      set D5.W = 4 (long)
   end if
   add ' ' to the op-code buffer
   copy D7.W to D0.W
   call the effective address routine (preserves D0)
   add ',A' to the op-code buffer
   call the dest register routine.
   Done.
else (CMP.size <ea> ,Dn and EOR Dn, <ea>)
   D0.W = 0 1 or 2 for byte, word or long
   Shift D0.W left by 1 place (0 2 or 4)
   if D0.W = 0
      set it to 1 (byte)
   end if
   Copy D0.W to D5.W (correct size specifier)
   Copy D7.W to D0.W again.
   if bit 8 of D7 is set
      copy 'EOR' to the op-code buffer
   else
      copy 'CMP' to the op-code buffer
   end if
   Call the size decoding routine (preserves D0, sets D5)
   If bit 8 of D7 is set (EOR Dn, <ea>)
      add 'D' to the op-code buffer
      call the dest register routine (preserves D0)
      add ',' to op-code buffer
   end if
   Call the effective address routine (D5 already set)
   If bit 8 of D7 is clear (CMP <ea> ,Dn)
      add ',D' to the op-code buffer
      call the dest register routine (preserves D0)
   end if
   Done.
end if
```

Type 26

This family is another that uses the condition code sub-routine. The 'Scc' instruction is the only one in this family.

```
Example 'Scc <ea>'
Call the condition code routine (preserves D0).
Add ' ' to the op-code buffer.
Call the effective address routine.
Done.
```

Type 27

This is the MOVEM instruction which has a nasty sting in its tail. According to the Motorola documentation, there is a word of data following the op-code which defines the registers that are being MOVEM'd. For all destinations EXCEPT the address register with pre-decrement, MOVEM reg_list,-(An), the word is in A7->A0 D7->D0 order (starting at bit 15) while for Address register with pre-decrement only, the order is D0->D7 A0->A7 starting at bit 15. We need to check for the latter case and reverse the bits in the register list word to be A7->A0 D7->D0 as well. This allows us to use a sub-routine to extract the appropriate bits & convert to register names. (And THAT is the absolute sub-routine from hell - see the next installment!)

If we are doing registers to memory (Bit 10 of the op-code is clear) then we cannot use post increment but we can use pre-decrement addressing. This implies that we need to reverse the order of the bits in the register list word to keep it in step with the other modes. (This is what Andy Pennell does as well) but what Andy doesn't do, that Motorola say should be done, is to test the mode bits in the op-code to determine if pre-decrement is being used or not. If the mode bits are '100' then we should reverse the mask word, otherwise we won't.

```
Example 'MOVEM.size register_list, <ea>' or 'MOVEM.size
<ea> ,register_list'
If bit 6 of D7 is set
   add '.L' to the op-code buffer
   set D5.W to 4 (Long)
else
   add '.W' to the op-code buffer
   set D5.W to 2 (Word)
end if

Get the word at (A6)+ and save it on the stack - the
register list mask word
if bit 10 of D7 is set (memory to register)
   Call the effective address routine (preserves D0)
   add ',' to the op-code buffer.
   get the saved reglist mask word from the stack
   call the address register list routine with the high byte
 of the mask
   if the data mask and address mask bytes are non-zero
      add '/' to the op-code buffer
   end if
   call the data register list routine with the low byte of
the mask
else (register to memory)
   get the saved reglist mask word from the stack
   if this is pre-decrment mode - bits 5,4,3 of the op-code
 word = 1,0,0
      The mask word is currently D0->D7 A0->A7 we need
     A7->A0 D7->D0 instead
      so swap bit 0 to bit 15, bit 1 to bit 14 ... (reverse the
    bit order)
   endif
   call the data register list routine with the low byte of
the mask
   if the data mask and address mask bytes are non-zero
      add '/' to the op-code buffer
   end if
   Call the address register list routine with the high byte
of the mask
   add ',' to the op-code buffer
   call the effective address routine
end if
Done.
```

Type 28

The type 28 instructions are a lot simpler than those we have just slaved over above.

```
Example 'MOVEA.size ‹ea› ,An'
If bit 12 of D7 is set
   add '.W' to the op-code buffer
   set D5.W to 2 (word)
else
   add '.L' to the op-code buffer
   set d5.W to 4 (long)
end if
Call the effective address routine (preserves D0).
Add ',A' to the op-code buffer.
Call the dest register routine.
Done.
```

Type 29

This one is quite simple, but looks nasty. It has two effective addresses - one for the source and one for the destination, and a non-standard size mode which we have to take care of. Having said that, it is really quite simple to decode as most of the work is handled by the effective address sub-routine.

```
Example 'MOVE.size ‹ea› , ‹ea›'
Mask out all but bits 12 and 13 of D0.W.
Shift D0.W right 12 bits.
if D0 = 1 (Byte)
   add '.B' to the op-code buffer
   set D5.W to 1
else
   if D0 = 2 (Long)
      add '.L' to the op-code buffer
      set D5.W to 4
   else (word)
      add '.W' to the op-code buffer
      set D5.W to 2
   end if
end if
Copy D7.W to D0.W.
Call the effective address routine (preserves D0)
Add ',' to the op-code buffer.
Shift D0.W 6 bits right to put the destination effective
address bits into bits 0-5 ready to decode again.
Call the effective address routine (preserves D0)
Done.
```

Type 30

Another simple instruction to decode.

```
Example 'ADDX.size -(Ax),-(Ay)' or 'ADDX.size Dx,Dy'
Call size decoding subroutine. (Also adds a space)
If bit 3 of D7 is set
   add '-(A' to the op-code buffer
else
   add 'D' to the op-code buffer
end if
Call the source register routine. Preserves D0.
If bit 3 of D7 is set
   add ')' to the op-code buffer
   add ',-(A' to the op-code buffer
else
   add ',D' to the op-code buffer
end if
Call the dest register routine.
If bit 3 of D7 is set
   add ')' to the op-code buffer
end if
Done.
```

Type 31

This is the catch-all which is called when the data we are decoding doesn't match any of the defined types.

```
Copy D7.W to D4.W.
Convert D4.W to hex and add it to the op-code buffer.
Done.
```

Next issue, we will have a look at the sub-routines.

An in-depth look on wxqt2 (Win)

Phoebus R. Dokos

I came up with the idea of a wxqt2 article when preparing the Type I versions of the Fontbuster CD True Type fonts for Dilwyn. The whole process just wouldn't be possible if it weren't for wxqt2 / qxltool. However, in the process of making that CD (more than 5000 files) I run across a series of limitations of the programme which I felt ought to be mentioned. This article deals with the best ways of installing wxqt2, do's and don'ts and well some tricks I came up withto help you if you get stuck.

What is wxqt2?

Wxqt2 is a front end to the excellent qxltool and qltools programmes by Jonathan Hudson. It facilitates the trasfer of ql files to/from the native file system and *.win files. It is VERY fast and for people like me who like ease of use, it overcomes the "unix crypticness" of qxltool.

Installation

After long hours of fooling around with it I came to the following conclusions about the best way of installing it. Most of these observations (I have to admit) are not too "scientific" but they work! So here they are:

1. Use as "shallow" as possible directories: WXQT2 starts faster and crashes less often when installed to the first level of the tree (e.g. c:\qxltools)
2. Put the WX22_0.DLL file in the same directory as WXQT2!. Although theoretically this should be put in the SYSTEM directory under Windows, I've

found that WXQT2 takes a "visible" additional amount of time to start as compared to having both files in the same directory. (I think it's purely Windows' fault BTW)

3. Add WXQT2's path to the path statement in the autoexec.bat file (for Win9x users this can be done either by using a text editor or by invoking SYSEDIT from Start›Run - for Win NT users this can be done through the Control Panel›System. Don't forget also to edit autoexec.nt in your c:\winnt\system32 directory). For reasons unknown to me, this seemed to correct many startup freezing problems) 4. Put all the programs in the same directory. e.g. wxqt2.exe should be together with qxltool.exe. It can run if you have them in a separate directory but again it solves many "freezing" problems during its startup.

Do's and don'ts

Below are some things that do not pertain necessarily to the installation but also have to do (IMHO) with MS-DOS / Windows limitations and which prevent wxqt2 to operate normally at all times.

A. DON'Ts

1. DO NOT use 4DOS on your startup if you are to use wxqt2 (wxqt2 REFUSES to start whenever 4DOS is loaded in config.sys with the "INSTALL=" directive)
2. DO NOT format QXL.WIN files from within WXQT2 if you are to make them larger than 98Megs (This came up after lots of testing). (See DO's for the remedy)
3. DO NOT try to access native directories with more than 1024 files (See DO's for the remedy). WXQT2 FREEZES worse than Windows on a clear day if you do so! For more than 512 (but less than 1024 files) be advised that WXQT2 will start with a significant delay (sometimes up to two minutes... also see DO's)
4. DO NOT change settings on WXQT2 without quitting the program for each configuration change every time (also see DO's for remedy). WXQT2 most likely will freeze. This is because instead of using an .INI file, WXQT2 uses Windows registry (see Picture) which is read once every startup of the program.
5. DO NOT try to restart WXQT2 after running it a couple of times when it froze at startup (see DO's for remedies... two ways do achieve this)
6. DO NOT attempt to re-install WXQT2 after a persistant freeze without cleaning up the registry first (see DO's for remedy)
7. Try NOT to run a lot of programs concurrently with WXQT2. Most likely you'll end up seeing the blank window of death(!) again :-)

B. DOs

1. For QXL.WIN files larger than 98Megs (practically for everything larger than 8 Megs without delay) use QXLTOOL directly by invoking the command "qxltool -w /dir/nameofqxl.win Size_In_Megs Volume_Name. It NEVER EVER hangs and it is faster. My guess for this problem is that it has to do something with the File Access mechanism of Windows. (It works fine in Win NT btw)
2. DO break down your huge directories to smaller ones. e.g. For the fontbuster CD I broke down files into directories according to the starting letter of the name. WXQT2 seems to work just fine even for larger amounts (›512 files) if this approach is used)
3. DO wait upon WXQT2's startup whenever the last access directory is a big one. Don't try to access Windows scheduler or any other program during this time because WXQT2 WILL freeze.
4. DO one of the following in an event of persistent freezing of WXQT2 (also see no. 5 below on how to avoid it alltogether).

a. You can access the registry by invoking the command Start›Run›Regedit. Locate the Key under

HKEY_CURRENT_USER-›
Software-›dld.org-›wxqt2

and erase both the value of the Device and QXLDev entries... NOT THE KEYS but their values. This will make your WQXT2 start up happily again.

b. If you are impatient just go and delete the last qxl.win file you tried to access and/or rename the last directory you tried to access (in case this wasn't one of "c:\" or "c:\windows" -and subdirs of windows- This takes care of either wxqt2 can't access the qxl.win file because it's broken or because the native dir is larger than what it can handle. WXQT2 will complain upon startup but WILL start!

5. To avoid this alltogether try to do one step at a time when configuring it. The best way to do it is the following (gives the least problems).

a. First you deselect the panel check marks alltogether.
b. You quit WXQT2. Then you restart it.
c. You select the native directory (without checking the panel check box)
d. repeat step b.
e. You select the QXL.WIN file (again without checking the panel check box)
f. repeat step b.
g. You select the panel check box for native directory. If everything works then proceed

with step h. If it freezes, that means that you need to work on the structure of the directory (break it down to more subdirs)

h. repeat step b.

i. You select the panel check box for the qxl.win file. If it works you're done! If it doesn't you know something is wrong with your qxl.win file. This procedure ensures that your last saved configuration is a usable one and you don't have to mess with the registry! (It appears that wxqt2 saves the settings upon exiting)

6. If you were unpatient and did not follow 5.b and 4 in the beginning but first tried to start it a couple of times, or if WXQT2 froze in the middle of doing something and you DO NOT want to restart, DO the following: Press CTRL-ALT-DEL once. Select how many instances of WINOLDAP are running and press End Task. Once the tasks are terminated and you did the registry clean up etc. then you can run WXQT2 again. Else just DO a Windows restart (only if nothing is wrong with WXQT2's configuration).

7. If everything else fails DO buy a QXL 2 and get rid of Windows 3:-)

Conclusions

WQXT2 is a very powerful and easy to use programme once you start it knowing its configuration limitations. Once you work around them (see above) it is faster than many even native Windows file managers (or even Explorer) on managing your files. In future versions though, I would like to see the following corrected:

1. Use of a INI file instead of a Registry Entry
2. More features in the configuration (like location of the help file) which is needed to be in c:\ (or / as it is called within WQXT2)
3. A help file (allthough I am not sure if I misplaced the file somewhere...)
4. An integration of qxltool and qltools with WQXT2. This way limitations of the multiple layers (WQXT2›WIN›QXLTOOL›DOS) can be overcome.
5. A DOS only version of WQXT2 for us mere we-want-everything-the-easy-way-mortals :-)

Acknowledgements

Many thanks to the following people. Without their help this article couldn't be a reality. Jonathan Hudson for giving me a reason to write :-) Thanks for this invaluable set of tools! John Hall for sending me the programs since for reasons known only to GOD I couldn't download it (a comment on everybody that tried to help me do so... I DO NOT USE a PROXY!!! ;-) Dilwyn Jones for his insight and for being the cause of discovering the aforementioned about WQXT2 (without you there wouldn't be a reason to... I hope everyone will soon enjoy the fonts!) Darren Branagh just for being a good sport! :-) and finally my infant daughter -who for reasons only known to Sir Clive remained silent while I was typing (probably had fun watching me)- and my wife for putting up with me cursing at my PeeCee everytime I was trying to copy the fonts to the .WIN files (and also bearing with me during the actual conversion)

For questions and/or comments about this article I can be reached at **dokos@adelphia.net** For US readers you can call me through DialPad (for Free) at 1-724-464-0199

The 9th Italian QL meeting

by D.Santachiara

The 9th Italian Sinclair QL meeting was held on Sunday 8th of October in Reggio Emilia (Italy). After four years it has been the occasion in order to meet old and new friends, Italians and not. The foreign exhibitors who participated were Jochen Merz (JMS Software), Tony Firshman (TF Services) and Roy Wood (Q-Branch).

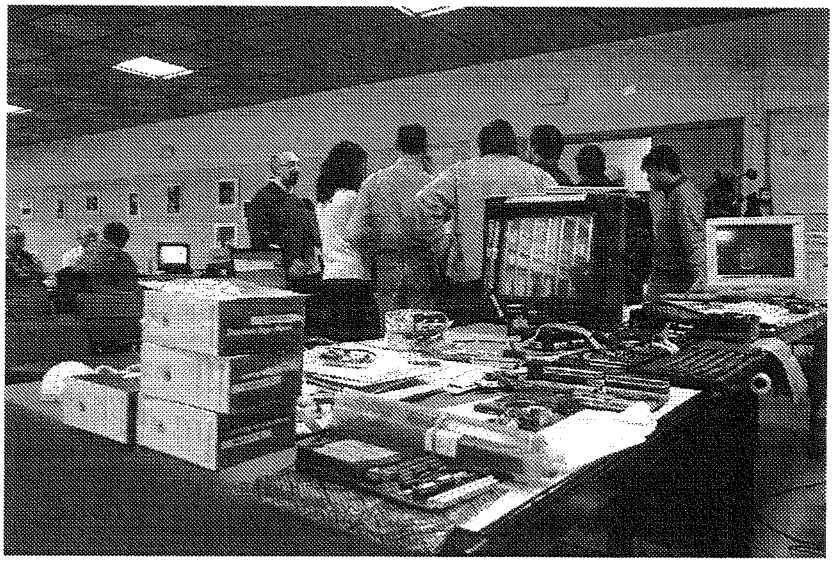
Nineteen Italian QL users attended the meeting. This is not a good figure especially because some "historic" QL Italian friends did not join us. We really put a big effort into trying to contact and invite as many users as possible but the feedback was not too satisfying.

After our foreign friends left Reggio Emilia around 4 p.m, we had an interesting discussion about the actions we should take in order to reverse this trend. One of the main reasons we found is the lack of communication and co-ordination between all the Italian users. For this reason one of the priorities that we have given ourselves for the future, is to create an "infrastructure" in order to keep connected and informed Italian users. For this task "the net" can help us a lot: in 1996 the email was still limited to an "elite" of persons, while right not now it is an instrument of widespread use. It is necessary therefore to take advantage of this technology in order to organise and keep informed the QL Italian community.

This is the reason why I have created an Email Italian bulletin (sinclair-info on **http://www.onelist.com**), which will periodically inform Italian users of news in the QL scene. So far this mailing list counts 65 users. Of course we will try to ship the newsletter via standard mail for people who do not have internet access.

On the other hand we will try to put on the web all the files collected on Italian Fidonet BBS: Ergon BBS (which has been recently closed down) and Qitaly BBS (which closed down more or less two years ago).

Finally Giorgio Garabello has prepared the resurrection of the former Italian QL disk magazine Qitaly. This completely new disk magazine is called QL Magazine and is worth having if you are able to read the Italian language. It can be downloaded from **http://utenti.tripod.it/Sinclair/qlm.htm** and can be read even from a 128Kb QL (640 Kb needed to view pictures). It can also be seen from PC by means of any html viewer (so it can be virtually viewed on almost every computer platform) and ZX Spectrum thanks to a separate utility that can be found on Giorgio's web site.



I started from the end, but now I'll return to the meeting. As we did in the past, the day before the meeting we had dinner together and this was really amusing. Unfortunately, unlike previous times, the only Italian participant has been the undersigned. The only difficulty was to find in the fattening "emiliana cousine" some vegetarian dishes for our guests. Tony Firhsman delighted us talking about his experience with insurance companies after his crash during the trip to Croatia in 1998 (see previous QL Today for full report). We talked about former QL developers like Stuart Honeyball alias Miracle Systems, Freddy Vaccha and many others. It was really great fun.

On Sunday morning I had just time to watch on TV the victorious arrival of Micheal Schumacher in the Japanese F1 grand prix. (Ferrari won the F1 championship after 21 years!). At 9:15 I opened the room. Jochen and the other guys arrived at about 10, with a slight delay because of a "misunderstanding" with the manager of the hotel who tried to apply higher rates than the ones reported in the rooms and inside the lounge. The first QL users who arrived there were my friends Adelchi Moscardini and Roberto Baciglieri (Adelchi had a heart-operation in hospital the week before!). Unfortunately the room sadly remained empty till 11 a.m. when the other Italian friends not coming from the Reggio Emilia area finally started to join us. At noon we started the usual discussions and presentations.

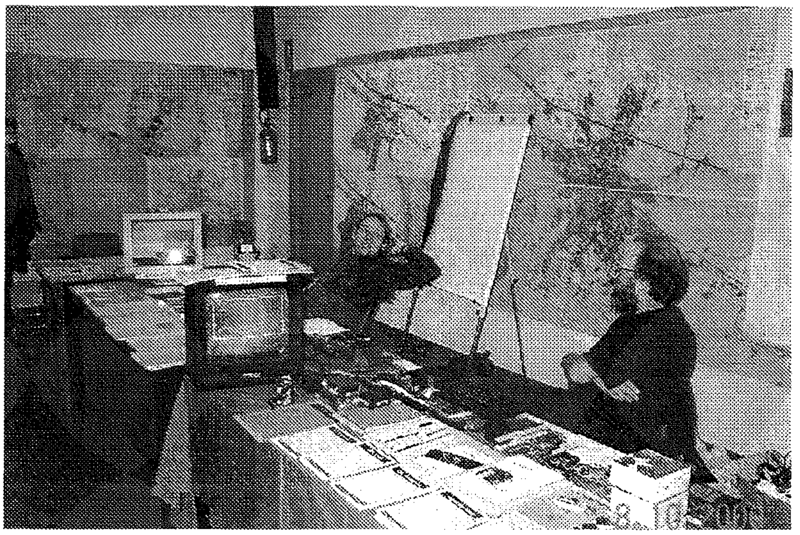
We started talking with Tony Firshman (**http://www.firshman.demon.co.uk/**) who introduced all his hardware products. Minerva MKII allows interfacing with the external world through some I/O lines. I2C are interface cards that allow to control electrical equipment (Tony did the demonstration with a Lego helicopter). Super-Hermes allows to connect to the standard QL a PC keyboard and a PC mouse. It has a fast serial port (up to 57600 baud), and it fixes the bugs of QL serial ports. A light version is available only with PC keyboard interface (plus standard serial port fixing). M-Plane is a backplane for the QL that allows connecting all the QL peripherals, from the standard QL to Aurora, Qubide, Gold Card, Super Gold Card etc. Romdisq is a small card that connects in the ROM port of the QL or in a special M-Plane slot. Romdisq has a static ram static (from 2Mb to 8Mb) that is seen from the QL like a standard ramdisk. Being a static ram it maintains the data even when you switch off your equipment, hence Romdisq becomes a small hard disk (seen the dimensions of QL files). Tony came from England by plane with a Romdisq, Minerva, SuperGoldCard and SuperHermes. I lent him for the meeting a QL and monitor and all was ready for his demonstrations without he had to carry a hard disk: all software was on Romdisq!

Jochen Merz (**http://www.j-m-s.com/smsq**) presented all his vast software production. The most noticeable news was the release of QPC2 version 2. This version (at the moment still in beta version but the final release will be available soon), supports 65536 colours (also through SuperBasic commands) and allows to use QPC2 in a Windows' window (version 1 takes all the screen). I really hope that it will be possible to read DOS files via a dosx_ device like (though in a different way) Daniele TerdinaÆs QemuLator does. The emulation kernel has been further streamlined and is approximately 20% faster than version 1.

The update from version 1 costs around 40 euros. I strongly believe that QPC2 is the future of the QL: it is supported in an excellent way by Marcel Kilgus (**http://www.deuschle.de/qpc**), it has the advantage to benefit from the great steps done in the PC world in terms of performances. You upgrade your system from a Pentium 200 to a Pentium III 800? QPC will go at least 6 times faster and you won't spend anything on the QL side. You can have a portable QL or a workstation QL: just install it on your notebook or your workstation PC. It is easy to network the QL in an extremely efficient way taking advantage of the Windows network (e.g. selecting the local QXL.WIN like win1_ and the QXL.WIN on another computer like WIN2). I think QPC2 is one of the most important programs ever produced for the QL.

Jochen also introduced version 2.98 of SMSQ/E for QXL. This provides 65536 colours like QPC2 v2. With the respect to the previous version the I/O speed has been improved. I've got and tried it but I must say that on my 4 Mb QXL the 65536 colours configuration is almost of no use because of lack of memory. Less than 1.5 Mb are available without loading extensions. After opening two or three windows the memory has gone. I wasn't even able to view at 65k colours a jpeg image with Photon because of insufficient memory. So if you are interested in the colours I would suggest the upgrade only to people who have an 8 Mb QXL. Said this please note that due to the large memory use for screen updating, even with the improved I/O speed the screen update redrawing appears sluggish. Instead it is now very good with the original 4 colours.

Qbranch (**ttp://www.qbranch.demon.co.uk/**) - Roy Wood - presented many software and hardware pieces that he distributes. After the withdraw of Qubbesoft, DJC, Miracle Systems etc. Qbranch is now the reference for software and hardware in the UK. You can find reconditioned / 2nd hand hardware (Gold Card, Aurora, Super Gold Card, QXL...) and new ones like Q40. Q40 (http://www.q40.de/) was probably the most awaited hardware at the Italian meeting: it is the real successor of the QL. It is based on a Motorola 68040 40 MHz CPU. It has slots for SIMM (up to 16 Mb), two ISA slots - one of which is used for the i/o card in order to connect the hard disk, serial ports, mouse and keyboard. The graphic card (1Mb memory) allows to get 1024x512 pixels resolution with 65536 colors. On Q40 you can run 3 different operating systems: QDOS classic (free, derived from Amiga QDOS û it has not all the SMSQ/E new features but it allows to get the extended resolutions with 4 colors), SMSQ/E (available for an extra price of 50 euro - it has all SMSQ/E features plus 65536 colours support) and LINUX. During the meeting we had a demonstration of SMSQ/E and LINUX with X-WINDOWS.



One of the most interesting things about Q40 is the capability to connect virtually every PC card with ISA interface (CD ROM, CD ROM burners, network cards...). Unfortunately while under LINUX the drivers already exist (in fact such cards work on Q40 with Linux), on the QL the drivers do not exist. That is why we need volunteers to write them in order to support new cards on Q40.

I mentioned before QemuLator. This is a shareware QL emulator for Windows written by an Italian guy: Daniele Terdina. Daniele was not able to attend the meeting because he is now working in the U.S at Microsoft. So I did myself some demonstrations of QemuLator at the meeting. I really like it because it can emulate the original JM, JS, MGx and Minerva QL systems allowing very old games or badly written software to run as it supports various ram configuration (128Kb to 16 Mb). It runs in a Windows (9x, ME, NT, 2000) window but it can use the full screen thanks to a new feature implemented in version - extended resolutions are not supported though. I think that QemuLator is completely complementary to QPC2 and is worth having. The latter is a real professional product with all the great features of SMSQ/E. QemuLator in this respect is a step behind QPC2 but it has other peculiarities such as the capability of emulating original QL systems (it has even a feature of simulating the original QL speed to slow down games on fast PCs). If you have QPC2 I would suggest to give a try to QemuLator. Give a look at Daniele's web site:

http://www.geocities.com/SiliconValley/Heights/1296/index.html

Last but not least Beginner's club from Vercelly (Andrea and Paolo Carpi - **http://www.geocities.com/siliconvalley/lab/5011**) came with their QLs with screen background of Japanese "Anime" and showed their good freeware software produced so far.

CONCLUSIONS: It was really great fun to meet again QL friends after so much time. Let's hope the next meeting can have a better attendance. You may find the full photo report on my Internet web site (**http://get.to/ergon**).

Hard Questions and Hard Disks

Geoff Wicks

The Dutch national newspaper "Trouw" recently published on article on "The Leaking Laptop":

"A laptop computer is not only easy to use, but also easy to steal. Goodbye years of work. Goodbye confidential patient details. At the very least a laptop user should think of making a backup."

The article quoted a representative of a German company, Utimaco, which specialises in computer security:

"A new client comes in with a hard disk crash, which contained his life's work, a research thesis on which he has worked for 8 years. Or the director of a cleaning company who was deeply shocked because he had been offered a stolen laptop containing details of his competitor."

"We notice that noticeably more laptops are stolen from employees of a bank or multinational than from the representatives of a beach ball company or similar. There is deliberate targeting, often from the same person several times a year. This can be as much due to the clumsiness of the person concerned as the value of the material he has with him. I know one multinational that forbids the use of a special laptop case; just put it in an Albert Heijn bag." (Albert Heijn is the largest supermarket chain in the Netherlands.)

A police officer at the Expertise Centre for Data Protection reacted:

"How do I carry mine? In a good case. Albert Heijn bag? No, I didn't know that, it's a good tip."

Many of the security problems of PC users pass we QL users by. There is not much profit to be made from stealing and fencing our black boxes, but many of us take our QPC laptops to shows. Mine goes in a good laptop case, which then goes into a tatty old bag. Nevertheless, the consequences of losing your hard disk, whether through theft or crash, also apply to QL users, as I have twice discovered during this year.

My 'workhorse' QL is a QXL2 card in a 286 PC. It has done good service, but age is catching up on it, and within a few months I have had two hard disk crashes. The hard disk is only 120 Mb, and as a third of it is the QXL_WIN file any crash is almost certain to corrupt the QL hard disk.

The first crash affected only the QXL_WIN file, and, after my initial panic, I reformatted and restored the QL hard disk to its original state much quicker than I expected. The main disaster was no backup of my Just Words! Line Design files, and, to complicate matters further, the day after the crash Roy Wood emailed me with an urgent request for more stock and labels.

I wrote off the first crash as a bit of bad luck, but the second crash made me think. This one affected the MS-DOS operating system, and I feared for a while I would have to reformat the disk and do a complete reinstallation of both the PC and QL sections. It was the last thing I wanted as I was busily preparing for the Portsmouth QL2000 show. Fortunately I was able to restart the computer from floppy disk and use Norton to examine and repair the hard disk. The damage was less than I expected, but even so the repair has not been perfect and this is affecting the performance of the QXL.

These experiences made me realise how little we know of QL hard disk use. Most of what has been published has concerned the technical side of QUBIBE and the hard disks it would support. What we do not know is how QL users use their hard disks. There are three areas of interest, the size and structure of a hard disk; maintenance and back up; and finally personal information and security.

Size and Structurl

3 - 5 years ago QL users frequently made comparisons between their PC and QL hard disks. They said they had to compress their files to get everything on their PC hard disk, but that their 20Mb QL hard disk was only half full. Is this still so? If we are using a QXL or QPC, how much space do we allocate for the QLX_WIN? What influences our decision? Do we partition our PC hard disks into say C and D drives to create a separate section for the QXL_WIN? Would there be any advantages or disadvantages in doing this?

In the QL world we pride ourselves on the absence of "bloatware", but will this always be so? If you do a full installation of ProWesS and Line-Design with all the fonts and clip art, a 20 Mb QL hard disk begins to look very small. And if we look into the future, do the new colour drivers mean we shall be storing more images on our hard disks? Or even further into the future, what effect will QL internet access have on the size of our hard disks?

What do we put on our hard disks? Just the programs we use regularly, or everything we come across? Do we still run some, perhaps rarely used programs, from floppy? When we install software do we do a full installation including documentation and example files, or do we select only the essential files?

How do we determine the structure of the disk? How many subdirectories and levels of subdirectories do we use? Do we use long or short subdirectory names? I tend to have a separate subdirectory for each program, but there are some exceptions. All my QTYP files are in the Text87 subdirectory. It is flattering to know some of my customers have a separate Just Words! subdirectory, but I do not. I would find the number of files in the directory confusing.

Maintenance and Back-Up

The PC has numerous programs and utilities for hard disk maintenance, including disk scanning, diagnosis, defragmentation, back up and compression. The QL has scarcely any. Many QL users argue they are unnecessary because of the absence of "bloatware". But will this always be so? If QL software can keep up with current hardware and operating system improvements, it is going to become more complicated. More colours, more graphics, more images, scaleable fonts and internet access will mean more complicated programs.

Look at my programs. Version 2 of QL-Thesaurus was 60K, version 3 84K and version 4

90K. Version 2 of Style-Check was 54K, version 3 104K. That is partly the difference between pointer and non-pointer programs. Is the time coming when we shall need QL disk scanning and defragmentation software?

What QL hard disk maintenance software already exists? The only commercial products I know of are Mark Knight's back up program and two utilities, DRVCHK and DRVLINK, that come with SMSQ-E. The last two utilities come with health warnings that are not quite as extreme as those on cigarette packets, but which make you think twice before using them. When I tried DRVLINK all it did was tell me some files had been corrupted, which I knew already.

Has anyone else used these utilities? Or other software for back up and maintenance? What were their experiences? Come to think of it, we know a QXL_WIN file is a standard DOS file, but we do we know how it is internally organised? Can the same disk maintenance/backup programs be used on a QXL_WIN as on a QUBIDE?

Now the controversial bit. We have all heard the often repeated advice that we must back up everything on our hard disk. This is not just complete nonsense, but almost an impossibility. Think how many floppies and how much time I would need to back up the 3.88GB hard disk of my laptop.

In practice few of us back up our entire hard disk. What we do is a selective backup. What criteria do we use when we decide what to back up and what not to back up? What problems do we come across?

I back up little of my QXL_WIN, because I have a hard disk on both my QXL machine and my laptop and they act as a back-up for one another. However, as a result of my two crashes I am wondering if a series of backup disks would also be sensible. One slight disadvantage of my policy is that I have to keep two hard disks up to date instead of one.

When my hard disk crashed I found the greatest problem was not re-installing the program files, but rewriting the boot file. I use only a few programs daily, and they were quickly reinstalled to give me a working QL. I reinstalled the others as and when I needed them. What I had not thought about were the slight differences between the boot programs. The laptop uses SMSQ-E and the QXL SMSQ, so the latter requires the installation of the pointer environment files. The laptop has a UK keyboard, but the QXL a Dutch (i.e. USA) keyboard. The laptop boot uses a disp_size command that is not present in SMSQ. ProWesS is not very stable on my QXL system and the installation is slow, so the boot gives the option of installing this. Needless to say, I have learnt from this and my boot is both backed up and printed out.

Personal Information and Security

I probably have a greater obsession with this than most people because for 17 years of my working life I worked in professions with a code of confidentiality. But even the professionals make their blunders.

Some years ago I took over a computer and discovered the highly secret address of every women's refuge in the Netherlands and the confidential minutes of one of them on the hard disk. Fortunately no harm was done as I could delete the files straight away, but why had the person concerned not deleted them herself? And had she realised that even deleted files can be recovered?

Because we cannot physically see the records on our hard disks, we tend to be more lax about confidential information than if they were paper records. What does your hard disk reveal about you? In the last year my QL has been used to type correspondence to a parliamentary committee and to senior politicians; letters to the tax and other authorities; and the papers of some quasi-legal proceedings. It has been used to update the data base of Just Words! clients, and to make spreadsheets detailing my income and finances and making financial projections.

This is information I never save to my hard disk. All output from the Psion or similar programs goes onto floppy. If some one gained access to your hard disk, how much would they find out about you? How do we QL users handle the confidential information on our hard disks?

Now Your Turn

In the early days of QL Today there were several articles describing reader's boot programs. I would now like to see similar articles on hard disk use. It is a topic that encompasses many aspects of QL use and levels of QL user. The hardware specialists could tell us more about the different types of interface and why some can access a CD-ROM and others not. The software experts can tell is more about the structure of a hard disk file and of hard disk maintenance programs. All hard disk users can tell us how they organise their own hard disk.

QL Today always needs material. Here is scope for several articles.

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets.

Cost.. £24

*************NEW***************

superHermes

A major hardware upgrade for the QL

All Hermes features (working serl/2 at 19200, independent baud rates/de-bounced keyboard/ keyclick) IBM AT kbd I/f // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software)	**£90** (£92/£93)
IBM AT UK layout Keyboard	**£11** (£13/£15)
Serial mouse	**£8** (£8.50/£9)
Capslock/scrollock LED	**£1** (£1.50/£1.50)
Keyboard or mouse lead	**£3** (£3.50/£3.50)
High speed serial (ser3) lead	**£4** (£4.50/£4.50)

Hermes available for £25 (£26/£27) **Working serl/2 and independent input, debounced keyboard.**

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead)..................**£53** (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if requd). Send disk plus SAE or two IRCs

MKI...£40 (£41/£43) **MKII...£65** (£66/£67)

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq	**£39** (£40/£41)
4mbytes RomDisq	**£65**(£66/£67)
8 mbytes RomDisq	**£98** (£99/£100)
Aurora adaptor	**£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/ Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost .... **£34** (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface **16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)**

2 amp (for 8 relays, small motors) .... **£40** (£43/£44)

4 amp total (for motors etc) .... **£45** (£48/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..**£25** (£28/£29)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required.........**£25** (£27/£28)

Analogue Interface **Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting**................**£30** (£31/£32)

Temp probe (-40°C to +125°C)..............**£10** (£10.50/£11)

Connector for four temp probes.............**£10** (£10.50/£11)

Data sheets...**£2** (£2.50/£3)

Control software & manual (for all I/F).......**£2** (£2.50/£3)

QL SPARES

Keyboard membrane	**£24** (£25/£26)
1377 PAL	**£3** (£3.50/£4)
Circuit diagrams	**£3** (£3.50/£4)
68008 cpu or 8049 IPC	**£8** (£8.50/£9)
8301/8302 or JM ROM or serial lead	**£10** (£10.50/£11)
Power supply (sea mail overseas)	**£12** (£17/£21)

Prices include postage and packing (Airmail where applicable). Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address/ postal order, Giro transfer (58 267 3909) or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list **13 Nov 00**

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254 Fax/BBS: 01442-828255
tony@firshman.demon.co.uk http://www.firshman.demon.co.uk

Searching for a String

John Sadler

You may have wondered what algorithmn is used by editors to search for a string in a text file. Normally one would check the first character and if it does not match try the next character in the text until a match is found. When a match is found, try the next characters in the string and text until it fails or succeeds. If it fails repeat with the character in the text after the first match was found. This means search looks at part of the text it has already searched and is wasteful in time. This routine avoids this problem by moving up the search string by the amount of letters which would match if the next character in the text was a successful match. Thus it only checks each character in the text once only. The enclosed programme demonstrates this algorithmn. The procedure getnxt creates the overlap array by getting the overlap f the previous one and checking how far it can be extended. For those who like a little challenge the algorithmn can be extended so that the checking of a match of the string is done as the string is typed in. This is the algorithmn used in the emacs editor. A solution for this version will be printed in the next issue.

```
100 CLS
110 REMark INPUT "Input string to be searched ";A$
120 REMark INPUT "Input search string ";B$
130 A$ = "xyxxyxyxyyxyxyxyxyyxyxyxxy"
140 B$ = "xyxyyxyxyxx"
150 n = LEN(A$)
160 m = LEN(B$)
165 REMark Array to hold the overlaps of search string
170 DIM nxt(m)
180 j = 1: i = 1
190 s = 0 : REMark index of succesful match
195 REMark create array of overlaps
200 getnxt nxt, B$
210 REPeat z
215  REMark exit if succesful match or end of teext
220  IF s <> 0 OR i > n THEN EXIT z
230  prtstrg i,j,A$,B$ : REMark update display
240  IF B$(j) = A$(i) THEN
245   REMark Characters match try next characters
250   i = i + 1
260   j = j + 1
270  ELSE
275   REMark find overlap of characters
280   j = nxt(j) + 1
290   IF j = 0 THEN
295    REMark No overlap set defaults
300    j = 1
310    i = i + 1
320   END IF
330  END IF
335  REMark check for successful match
340  IF j = m + 1 THEN
350   s = i - m
360  END IF
370 END REPeat z
380 IF s THEN
390  PRINT: PRINT "Index of match is "; s
395 ELSE
397  PRINT: PRINT "No match."
400 END IF
410 DEFine PROCedure getnxt(nxt, B$)
415 REMark create array of overlaps if match fails
420  LOCal i, j, z
425  REMark Set first two values
430  nxt(1) = -1
440  nxt(2) = 0
445  REMark Now calculate the rest
450  FOR i = 3 TO m
455   REMark Get the previous overlap
460   j = nxt(i - 1) + 1
470   REPeat z
480    IF j < 1 THEN EXIT z : REMark No overlap
485    REMark Exit if the overlap is remaning string
490    IF B$(i - 1) = B$(j) THEN EXIT z
500    j = nxt(j) + 1 : REMark Try the next one
510   END REPeat z
520   nxt(i) = j
530  END FOR i
540 END DEFine nxt
550 DEFine PROCedure prtstrg(i,j,x$,y$)
555  REMark updates display of text string and search
     string
560  AT 0,1: PRINT x$
570  FOR x = 1 TO i
580   AT 1,x: PRINT " ";
590   AT 2,x: PRINT " ";
600  END FOR x
610  FOR x = 1 TO i-j
620   AT 3,x: PRINT " ";
630  END FOR x
640  AT 1,i: PRINT x$(i)
650  AT 2,i: PRINT y$(j)
660  AT 3,i-j+1: PRINT y$
670  PAUSE 50
680 END DEFine
```

The Quanta Library CD-ROM

Darren D. Branagh

At the Quanta AGM in Manchester in April earlier this year, I was approached by Roy Brereton of QUANTA to produce a CD-ROM version of the entire Quanta Library, based on my previous successes with making CDs - (the main one at that point was the Dilwyn Jones QL Collection, a CD Full of all Dilwyns Commercial and Freeware Programs to date.) The Quanta CD was launched at the QL2000 show in Mid-October, and sold extremely well - actually, We sold out of all the copies I brought and I had to take over Dilwyns CD-RW machine on his PC to burn a pile more!! (Cheers, Dilwyn!)

What is it?

Basically, the Library CD-ROM is a 650Mb CD-R recorded in ISO 9960 format and containing the entire Quanta Library of Programs (formerly only available on floppy disk - the entire library was on approximately 200 disks) available in one single CD. Absolutely every program submitted to Quanta over the many years of its existence is included - several thousand files, and well over 700 Programs including Games, Utilities, File readers and converters, connectivity and connection software to other computers, Programming Suites, and Patches, Updates and Demos. On the CD itself you will find a few files, which are as follows:

QXL.WIN - The most important file, as its the one that contains all the entire library files and directories, more on this later. It is obviously al arge file, being 150Mb in size.

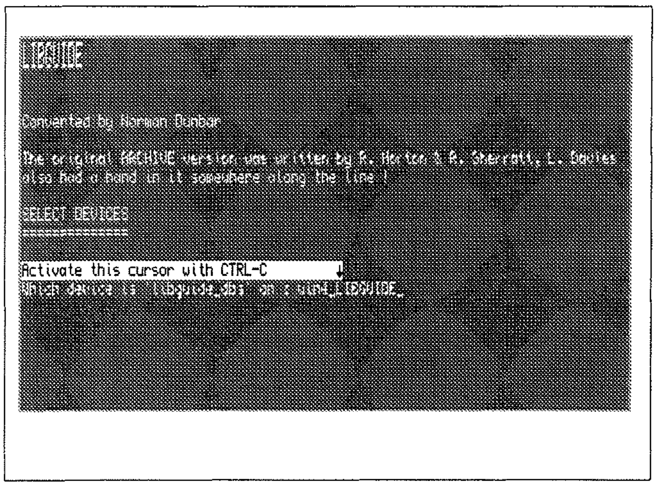
REAME.TXT, **README.DOC**, and **README.RTF** - Three versions of the same file, in ascii text, Windows Word format, and in Rich Text format. This file contains basic information and acknowledgements about the CD itself and its usage.

Also on the disk are a few additional freebie Zip files which may come in useful, and are rather difficult to get hold of without an internet connection:

QWE091.ZIP - This is a ZIP file (compressed) containing the useful QXL.WIN Explorer utility, useful for viewing the contents of a QXL.WIN file from Windows.

QXTOOLS.ZIP - Another ZIP file containing the QXL Tools utility. This was be useful in passing info from a PC to QL and reading media, etc.

STRIPPER.ZIP - Norman Dunbar's excellent utility for viewing and removing control codes from Quill DOCs on a Windows PC.



```
LIBGUIDE

Converted by Norman Dunbar

The original ARCHIVE version was written by R. Horton & R. Sherratt, L. Davies
also had a hand in it somewhere along the line !

SELECT DEVICES
==============

Activate this cursor with CTRL-C
Which device is libguide_bas on : win4_LIBGUIDE_
```

So, how does it work?

In order to use the Quanta Library CD-ROM (hereafter referred to as the CD) you will need a system that can adequately handle QXL.WIN files. Unfortunately, this limits the CDs use to the QL emulator systems mainly, though it can be used in a round about way on a Native "real" QL with a QubIDE fitted (see below for full details)

The Emulators that can handle the QXL.WIN format are currently:

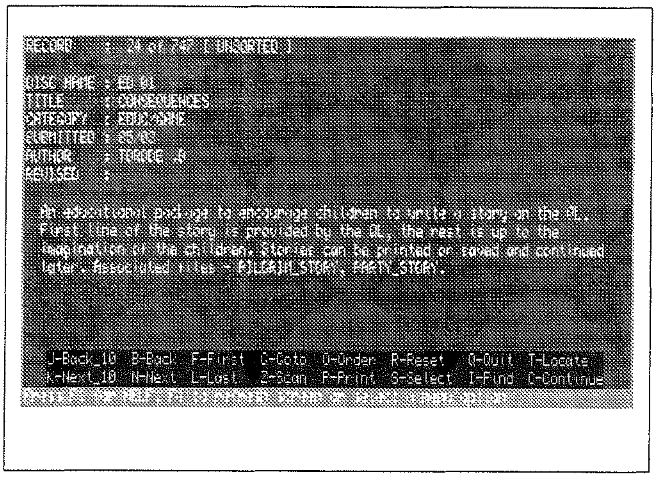
1. All versions of QPC, the Software Emulator for PCs by Marcel Kilgus. This includes both QPC1 and QPC2, and also the new QPC2 Version 2.
2. uQLx, the Software emulator for Linux/Unix machines.
3. Q-Emulator, The Software emulator for PCs by Daniele Terdina. Only the PC version of Q-Emulator can read QXL.WIN files at the time of writing, NOT the Apple Macintosh version.
4. The Miracle Systems QXL Card. This is a Hardware emulator for PCs, and this includes the faster 25Mhz QXL2 card also.
5. Any other systems or emulators capable of reading the contents of a QXL.WIN file.
6. Dave Walker says that the CD can now be used by people with a real QL and QubIDE - as long as they have the latest QubIDE ROM's and the latest version of Dave Walkers DiscOver software - the CD can be read on a native QL setup using QubIDE, then copy the QXL.WIN file across to a hard drive and the QL version of QL TOOLS by Jonathan hudson can then be used to read the files within the QXL.WIN file, direct from the copy on the hard drive.

As use of the CD itself differs depending on the machine you are using, I will not go into this in detail as its best to consult the manual for your particular QL emulator. Suffice to say that you simply configure your emualtor software to find a QXL.WIN file on the CD, and the emualtor should pick this up as a pre-defined WIN file when the emulator is running. Therefore, if you configure your CD to be WIN2_,2_, typ DIR WIN2_ will show the contents of the QXL.WINs howing all the library files in separate directories, just as on a normal QL.

Structure of the QXL.WIN file

The QXL.WIN file as I said earlier contains all the library files - viewed from DOS or Windows, it appears as a single file of 150Mb in size. However, from the QL side of things, when DIRd the entire Quanta Library is shown in directories.

The format of the CD keeps the old Quanta numbering System for the directory structure which



```
RECORD    : 24 of 747 [ UNSORTED ]

DISC NAME : ED 01
TITLE     : CONSEQUENCES
CATEGORY  : EDUC/GAME
SUBMITTED : 05/03
AUTHOR    : TORODE .A
REVISED   :

  An educational package to encourage children to write a story on the QL.
  First line of the story is provided by the QL, the rest is up to the
  imagination of the children. Stories can be printed or saved and continued
  later. Associated files - PILGRIM_STORY, PARTY_STORY.

   J-Back_10  B-Back  F-First  G-Goto  O-Order  R-Reset  Q-Quit  T-Locate
   K-Next_10  N-Next  L-Last   Z-Scan  P-Print  S-Select  I-Find  C-Continue
```

many of you with no doubt be quite familiar with, therefore the contents of directory LG01 (ie. The Library Guide) is the same as the floppy LG01. To save time, the contents of the CD have not been reconfigured in any way, they are the sameas when they were on floppy - therefore some configuring and changing of BOOT files etc. May be required. Of course, all files will work perfectly by copying the contents of each directory back out to a floppy and running from FLP1_. Also, all programs that are single file only and are stand alone can be EXECd or RUN directly from the CD itself. The Quanta Library Guide Database is included on the CD - in fact it is configured to BOOT this program on startup, and I suggest you use this to find the files you want on the CD. If a certain game you wish to use is listed by the Libguide as being found on GG02, it will be in the directory GG02 on the CD. (Simple, Eh?)

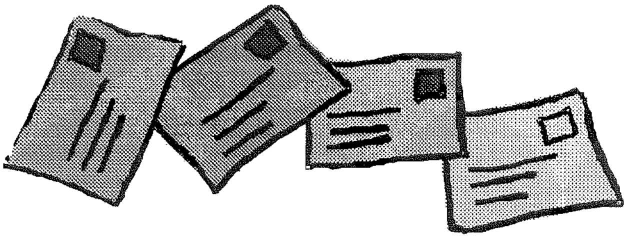
Sounds good, how do I buy a copy?

All orders for the CD should be made to the Head Librarian of Quanta (address inside the front cover of the Magazine) and should be despatched quite quickly. I will be supplying the CDs to Quanta and it is hoped an amount will always be held in stock by Quanta to avoid delays. The cost of the CD-ROM is a mere £10 Sterling (a Bargain, considering the amount of work that has gone into it -several months of spare time in my case!). If you want one, please get in touch as soon as possible to avoid delays as I expect demand will be high in the run up to Christmas.

Conclusion

So, there you go. The CD is now a reality (at last) and I'd like to thank Roy Brereton and the Quanta Committee for pushing me into doing it, and Jochen Merz for supplying me with loads of very inexpensive and Good Quality blank CD-Rs to make them with!! I hope this service will prove useful to many of you and you will find the CD useful and order one. Unfortunately, The CD (as I said earlier) is only of use to QL Emulator users at present. That, however, it not to say that this will always be the case. CD reading capability is feasible via QubIDE - all it takes is someone to write the drivers, and load them in at startup, as the QubIDE ROM's have no space left!! I also know that new versions of Dave Walkers Disc-Over software allow CD access via direct sector access (see above). So, as Xmas nears, I want to say that proper CD Accessibility for native QL hardware, ie. QubIDE, is top of my Wish List, especially as we now have a working TCP/IP Stack for email and internet access, it seems anything is possible in the QL World lately. Anybody want to write some real CD drivers for QubIDE to read QXL.WINs directly? I hope so.

Letter-Box



John Hall writes:

After chatting to Jochen at QL2000 about ESC/P2 printers and then reading the comments in QL Today, I've had a look at the Epson UK web site which gives the following information:

1) The Stylus 880 is ESC/P Raster only :-(

2) The Stylus 980 is ESC/P2 and IBM X24 :-)

So rather than rushing round trying to find an end-of-line Stylus Color 900, I'm going to wait until the box-shifters have stocked up with the Stylus Color 980!

John Mason wrote to us on the same subject.

***Jochen:** I have checked the EPSON website a few days ago. The 880 specification still says, scaleable font and ESC/P2, and the downloadable PDF prospect says the same. I was suspicious about the reduced number of fonts anyway, so I you are going to buy it, get it in writing that it supports ESC/P2. Whoever gets experience, please inform QL Today!*

I rushed and got a Stylus Color 900 - excellent printer, fast and compatible!

Ian Pizer writes:

Addendum to "Printing Accented Characters"

In my article in QL Today of Sept/Oct 2000 I mentioned using PARD to print from Text87 to avoid using a translation table. At the time of writing I could not find the origin of the instruction but it is mentioned in the Super Gold Card Supplement page II. PAR or PART is the default and will use an active translation table, but PARD avoids any translation. Thus this possibility is only valid for Super Gold Card.

Not quite: T and D are supported by any SMSQ/E system. Without SMSQ/E, you have to use TRA 0 (which means no translate for every output of every job). With SMSQ/E, use PARD only for the output which should not be translated, regardless of the TRA setting. - Editor

Hard is better

Jérôme Grimbert

Let's make it clear from the very start, this article is not a rant against QPC or any software emulation of a QDOSMSQ system. In fact, I'm glad such emulations do exist, even if I do not use them (yet?). No, the title of this article is the mere summary of two of my recent experiments. But first, let me describe my systems. Or rather only a part of them, I will not describe my old QLs nor their evolutions (with a Gold Card/ Super Gold Card, a Miracle Hard Disk, a Qubide interface, the Hermes/Super Hermes chips)... Oops, seems I started that anyway.

So back on the main track, the two systems involved in this experiment are a QXL2 in a PC and a Q40. In fact, the PC is also a host for a ZinZin95 and a Linux Operating system. The presence of a QXL2 in it in this experiment is mainly irrelevant. In fact this experiment resulted from my writing of a sprite editor for Q40: I soon wanted to convert all the nice icons from the other systems for the PE (in mode 33, 4 and 8, even if the resulting mode 4 or 8 is rather poor). So I wander in the various format when I was able to find some information about them. Two formats were soon omnipresent for the Linux system: xpm and pnm (which regroups ppm, pgm and pbm [not to be confuse with bmp]). And they have, for me at least, a great advantage: no compression (which means a simpler loader to write). For the ZinZin, I found a lot of collections on the Web, but positively no information on the format of a file. PNM was rather easy to decode, once the trouble for the optional comments was solved. XPM is a very versatile format, with nevertheless a big advantage over PNM: transparency is possible (just like with the PE sprite!). But XPM is so versatile about colour specification that it may require a long symbolic list for the colour name. Thus, I chose to let ImageMagick on the Linux to do some conversion from XPM to PPM. Alas, the transparency then disappeared and turned out as black. I later found a way to extract the transparency information into a separate PPM.

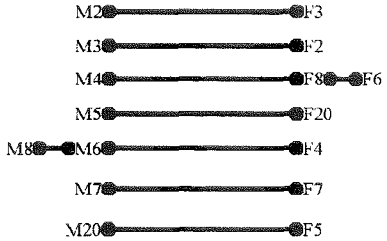
I would like to focus your attention on a little point: the pixel appearence between PC and QL is not the same. Pixel on PC are usually square on a 4x3 display (except for resolution like 320x200, 640x400 or 1268x1024) whereas on a QL (including a Q40), the pixel is rectangular. So to keep a circle aspect, when converting from the PC to QL, the X dimension should be extended by 1/3. But this has a side effect on perfect diagonal line which loose this perfection once extended.

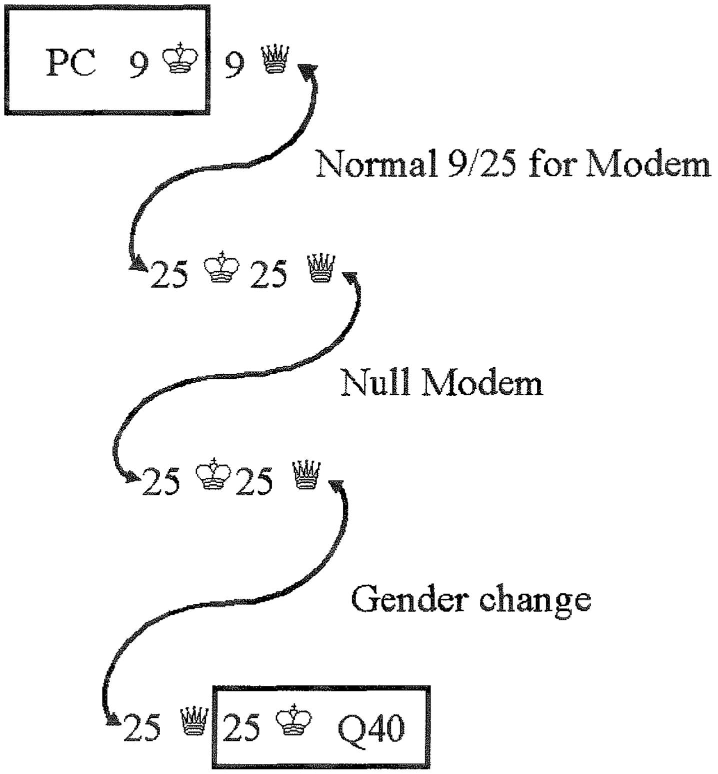
After a little find (a utility from Linux), I ended up with a 5 megabytes zip file containing a lot of PPM files (for colour and for mask, some extended and some not). As in the meantime I had written on my Q40 two converter, one which converts a PNM into a _pic file (in mode 33, so I can see it directly on the Q40), and one which could take upto two _pic files (one for colours, the other [optional] for the mask) and output an assembly file of a sprite. So, given:

- the very slow speed of the Q40 when reading a MS-DOS floppy (even more in my case, as I have a 32 Megs Q40)
- the fact that Linux does not support QDOS floppy
- my ignorance of how to split and recombine such a big zip file.

How much time did it take me to transfer this 5 megabytes zip file onto the Q40? Well, less than 10 minutes! How? Now that's interesting, as it takes already more than 15 minutes to copy a 1.44 Mega file on a MS-DOS floppy to my hard disk. The answer: hardware. The explanation: I have a serial link between the Q40 and the PC. And the Q40 serial card is rather good, it supports upto 115 200 baud (like modern PC)!

I will not restart the religion war that happened on the exact configuration of the serial cable (I provide two pictures of mine, but others have other settings, so as long as it works, it's ok with me).





For the software configuration on Linux, it's rather simple (let ttyS1 be my serial port on Linux (alias Com2 under ZinZin):

```
% stty 115200 </dev/ttyS1
% stty raw </dev/ttyS1
% stty ctsrts </dev/ttyS1
```

For the Q40, it's even simpler:

```
BAUD 115200
```

Then I prepare the Q40 for the reception:

```
copy ser1 to win4_a_uni_zip
```

And finally start the transmission with:

```
% cat uni.zip >/dev/ttyS1
```

Once the Linux reports the end of the command, I waited some more seconds before breaking (with Ctrl-Space) the copy on the Q40, and check the size of the file on both ends. Believe it or not, but unzip on the Q40 has no problem extracting the whole archive! So Hard is better than Soft (floppy disk were also called soft disk, in opposition to the hard disk!). And now for my second experiment. It involved only the Q40. Claus Graf had provided me (and the Web) with a color-picker window for the Q40, in its library qx0 (it's for C68). I incorporated it in some versions of my sprite editor on Q40, but I wanted more control of it. So I started writing my own (and bigger) one. As generating the whole disc of colours each time may have proven to be very cpu-intensive, I decided to display a big sprite (only about 300 x 200 !) in an application window. The big sprite is obviously the disc of colours, but I needed to generated its source before being able to use it. As I already had a converter from pic to sprite, I chose to write a very small program to generate a 300 x 200 pic. The main part of this image generation required some calculations with some floating numbers (double in fact, in C68), including a square root and an arc tangent. On my standard setting, the Q40 takes 1:47 (nearly two minutes) to do it. After a little lrespr of fpsave_bin (a nice code, really, which enable the use of the floating point unit, well that's what I've read), the same programs only takes 0:15 (yes, only 15 secondes). That's really an improvement. So here again, hard is better than soft!

Hints and Tips

QL Quill Devices

by David Denham

I admit it: I still use QL Quill. I suspect many of us are "closet" Quill users, finding the simplicity of use and friendliness keeping us using Quill despite competition from the Perfections, Text87s and Paragraphs of this world.

It can be quite useful to be able to print or save to or load from devices other than those for which Quill has been pre-configured. Of course, we can use the DEV or SUB devices to fool Quill into saving to or loading from hard disks and other devices, or even get Quill to print to a file.

I found by accident that if you precede the name of a device with an underscore, Quill seems to accept it. For example, if you tell Quill to load from _SER1 it will sit there patiently waiting for input from SER1. Not particularly useful I suppose. You can tell Quill to print to _PAR if you wish to print to a second printer on a Super Gold Card's parallel port - useful if your main printer is on SER1, and you have an old spare dot matrix printer lying around to be used as a "part-time" label printer for your business for example.

Recent versions of Quill will also let you access the network if you precede the network device name with an underscore.

As far as I know, this feature of QL Quill is not documented anywhere, and I have not tried it from the Xchange version of Quill. If anyone knows more about this feature (I assume it's a feature and not a bug!) please let us know via QL Today.

Q. I use QemuLator version 2. It claims to be able to handle sub-directories, but I am having difficulty accessing sub-directories.

A. *(from Daniele Terdina)* Make sure that the "Level 2 file system" option is checked in the Devices tab of the QL Configuration dialog, otherwise Q-emuLator emulates the original QL file system without sub-directories.

Oddities

Bruno Coativy

First of all, I would like to thank all those without whom QLToday would not be in existence.

I would also like to make a remark about Norman Dunbar's excellent series "Programming in Assembler". I happen to be very fond of assembler and I have seen something odd (sorry, Norman...) in instalment #8.

In part 8 (QLToday Volume 5, Issue 2, Page 40), Norman gives a piece of code which purpose is to tidy the maths stack with a string previously fetched (pointed to by the a1 register):

```
move.w    0(a6,a1.l),d0      (1)
addq.w    #3,d0              (2)
bclr      #0,d0              (3)
add.w     d0,a1              (4)
```

This code is fine in all cases but three: when the string fetched has a length equal to 32765, 32766 or 32767 caracters (i.e. the maximum allowed under QDOS or compatible operating systems).

In this latter case, for example, the d0 register will successively hold the following values (here, the "?" sign represents a four bits word of any value):

```
(1)  $????7FFF
(2)  $????8002
(3)  $????8002
(4)  $FFFF8002 (as seen by the a1 register)
```

So, whereas a positive offset of 32770 should have been added to a1, an offset of -32766 is added instead. I let you guess what happens from there on...

Obviously, the routine given on page 42 does not work as well for the same reason.

As can be seen, the problem simply lies in the fact that adding three to one of these three lengths gives a result that cannot be represented as a positive number in a 16 bits word. Hence the final, negative, offset. Sign extension can sometimes be dangerous!!!.

Here is an example of a small chunk of code that will work in all cases, without exception:

```
move.w    0(a6,a1.l),d0      (1)
ext.l     d0                 (2) <-- Sign extension
add.l     #3,d0              (3)
and.b     #%11111110,d0      (4)
add.l     d0,a1              (5) <-- No sign extn.
```

This time, the d0 register will hold the following values:

```
(1)  $????7FFF
(2)  $00007FFF
(3)  $00008002
(4)  $00008002
(5)  $00008002
```

An offset of $00008002 (32770 in decimal) - which is the correct value - is now added to a1 at step (5). It can be noted that the sign is explicitly extended at step (2) by means of an "ext.l" instruction.

This article illustrates - to a certain extent - why the assembler language is difficult. Basic rules are generally simple. What is really difficult is having them in mind all at the same time. Very difficult indeed. Yours truly knows why...

Normans comment:

Bruno has indeed found one of those 'gotchas' which we encounter from time to time in assembler programming. I have never come across this problem in all my years (!) of programming the 68000 processors, and indeed, my old favourite DJToolkit will fall foul of this bug should any very large strings ever be passed to the routines.

I have consulted my documentation on BV_CHRIX (allocate space on the maths stack) and it accepts the number of bytes required in a LONG word. IN other words, I have been doing it wrong all these years - and never (yet) fallen foul of the problem.

Thanks Bruno, for pointing out the error of my ways. That's another one to watch out for.

Assembler Homework Answer

Norman Dunbar

OK, I know I said I was not going to give the answers to the last lot of homework that you all did (didn't you ?) but Jochen begged me to do it, so here it is.

If you remember, I asked you to create a new procedure called PSI which did exactly the same as PSI_CLS that I wrote for you (and hopefully explained) in a recent article (I forget where we are because I keep having to split articles up to avoid hogging the entire magazine!). Anyway, the hints I gave you should have been enough to get you going, in case you have forgotten them, here they are again:

- update the definition table with details of the new procedure.
- in the proc's code, set D6.B to zero for PSI and 1 for PSI_CLS. Do this as the first instruction in both procedures.
- In the PSI procedure, simply set D6 and jump to the code in PSI_CLS.
- Just before doing the actual CLS part of PSI_CLS, check the value in D6.B and if zero, don't do the CLS simply BRA.S to error_exit instead.

Well, I said in less than 10 lines. I did it in exactly 7 additional lines INCLUDING the new definition for the PSI procedure. The following is the new code combining the old PSI_CLS and the new PSI procedures. Any lines you see below which have comments starting with ««« have been amended or inserted for the new PSI proc. Enjoy !

```
err_bp   equu   -15          ; Bad parameter error
```

```
err_bp   equ    -15         ; Bad parameter error
err_no   equ    -6          ; Channel not open
bv_chbas equ    $30         ; Offset to channel table
bv_chp   equ    $34         ; Offset to channel table end
bv_rip   equ    $58         ; Maths stack pointer

start    lea    define,a1  ; Pointer to the definition table
         move.w BP_INIT,a2 ; The vector we need to use (= $110)
         jsr    (a2)       ; Call the vectored routine
         rts               ; And return any errors back to SuperBasic

define   dc.w   2             ; ‹‹‹‹ 2 new procedures
         dc.w   psi_cls-*
         dc.b   7,'PSI_CLS'

         dc.w   psi-*         ; ‹‹‹‹
         dc.b   3,'PSI'       ; ‹‹‹‹
         dc.w   0             ; End of procedures

         dc.w   0             ; Number of functions
         dc.w   0             ; End of functions

psi      moveq  #0,d6         ; ‹‹‹‹ Flag for PSI
         bra.s  psi_both      ; ‹‹‹‹

psi_cls  moveq  #1,d6         ; ‹‹‹‹ Flag for PSI_CLS

psi_both move.l a5,d7         ; ‹‹‹‹ End of parameters
         sub.l  a3,d7         ; Minus start of parameters
         divu   #8,d7         ; How many parameters ?
         cmpi.w #3,d7         ; Are we defaulting channel id ?
         beq.s  hash_ok       ; yes, skip hash check

hash_check
         cmpi.w #4,d7         ; We better only have 4 parameters
         bne.s  error_bp      ; Oops !
         btst   #7,1(a6,a3.l) ; Is there a hash before parameter 1 ?
         beq.s  error_bp        ; No, we reject it then

hash_ok  move.w ca_gtint,a2     ; We want word integers
         jsr    (a2)            ; Fetch them all
         tst.l  d0              ; Did it work ?
         beq.s  got_ok          ; Yes it did
         rts                    ; Bale out with error code otherwise

got_ok   cmpi.w #4,d3           ; Were there 4 of them ?
         beq.s  got_4           ; Yes

         cmpi.w #3,d3           ; Maybe default channel in use
         beq.s  got_3           ; So that is ok too

error_bp moveq  #err_bp,d0      ; Bad Parameter error code
error_exit rts                  ; Bale out with error

got_4    move.w 0(a6,a1.l),d0 ; Get channel id
         bmi.s  error_bp        ; We don't like negative channels
         adda.l #2,a1           ; Tidy stack pointer
         bra.s  get_rest        ; Skip the default channel id bit

got_3    moveq  #1,d0           ; Default channel is #1

get_rest bsr    channel_id      ; Convert D0 to QDOS id in A0.L
         bne.s  error_exit      ; Bale out if errors

         move.w 0(a6,a1.l),d4 ; Paper in D4
         bmi.s  error_bp        ; Negative is bad news
         andi.w #$00ff,d4       ; Force range 0 - 255

         move.w 2(a6,a1.l),d5 ; Strip in D5
         bmi.s  error_bp        ; Negative is bad news
         andi.w #$00ff,d5       ; Force range 0 - 255

         move.w 4(a6,a1.l),d6 ; Ink in D6
         bmi.s  error_bp        ; Negative is bad news
         andi.w #$00ff,d6       ; Force range 0 - 255

         adda.l #6,a1           ; Tidy the stack

         moveq  #sd_setpa,d0    ; Paper trap code
         move.w d4,d1           ; Paper colour
         moveq  #-1,d3          ; Infinite timeout
*                               ; Channel id is still in A0
         trap   #3              ; Set the paper
         tst.l  d0              ; OK ?
         bne.s  error_exit      ; No bale out

         moveq  #sd_setst,d0    ; Strip trap code
         move.w d5,d1           ; Strip colour
         trap   #3              ; Set the strip
         tst.l  d0              ; OK ?
         bne.s  error_exit      ; No bale out

         moveq  #sd_setin,d0    ; Ink trap code
         move.w d6,d1           ; Ink colour
         trap   #3              ; Set the Ink
         tst.l  d0              ; Ok ?
         bne.s  error_exit      ; No bale out

*-------------------------------------------------------
* And finally, we can CLS the screen. You have seen this before in
* QLTdis.
*-------------------------------------------------------
         tst.b  d6              ; ‹‹‹‹ Do we need to CLS ?
         beq.s  error_exit      ; ‹‹‹‹ PSI called, no CLS required.

         moveq  #sd_clear,d0    ; CLS whole screen
         trap   #3              ; Do it
         bra.s  error_exit      ; All done

*-------------------------------------------------------
* This routine takes a SuperBasic channel number in D0 and converts
* it into a QDOS internal channel id in A0. If the channel is closed
* or not yet opened, the routine returns D0 = ERR_NO and A0 is
* invalid.  D0 will be zero if all is ok.
*-------------------------------------------------------
channel_id mulu #$28,d0         ; Offset into channel table

* Lots of code removed from here - to save space, you must leave
* the code alone !!
         rts                    ; Finished

ch_bad   moveq  #err_no,d0      ; Channel not open (-6)
         rts                    ; Bale out
```

RWAP SOFTWARE

Flashback SE v2.03(Upgrade) £5

This is the ultimate database program - extremely fast and flexible, easy to use, updated to cope with the latest versions of the QL operating system and still maintained. A report module is included to allow you to format output in any way, including mail-merge. Unfortunately, only available as an upgrade to the original version (Original still available from Sector Software).

QL Cash Trader v3.7 £5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT return, reports and analysis for audit trails and management decisions. Previously sold for over £100.*

QL Payroll v3.5 £5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of current tax year rules.

Q-Help v1.05	NEW VERSION	£10
Q-Index v1.04	NEW!!	£5

Q-Help: on-screen help for SuperBASIC commands, including TK2, Turbo Toolkit, SMSQ/E and PD toolkits. Can be used to add help to your own programs - simply produce ASCII text for each help page, add an index and Q-Help automatically cross-references and displays the links.
The PD toolkits referred to are available for £2.
Q-Index: The SuperBASIC index supplied with the Reference Manual - enter a topic such as 'screen resolution' and fnd out the commands which relate.

Sidewriter v1.08 £10

Produces landscape printouts of Easel/QSpread spreadsheets and output from QL Genealogist, as well as any other standard text file. You can specify the fonts to be used on the page. Works with all EPSON compatible printers, from 9 pin dot matrix up to inkjet printers. A most useful utility written by Dilwyn Jones - you know it must be easy to use.

ProForma ESC/P2 Drivers v1.01 £8

New improved printer drivers, providing up to 720 dpi printout in full colour from all programs written for use with ProWesS, such as LineDesign and Paragraph. Work on all Epson inkjet printers which support binary mode compression (740,850 and 900 models at least). 1440 dpi to follow.

QL Genealogist v3.26	£20
Genealogy For Windows	£50

Keep track of your family tree! Add individuals, with details of their parents and children, watch all of those links build up into a formal family tree layout. Text files and pictures may also be linked to individuals as well as notes and events, making this the perfect way to preserve the history of your family for future generations. QL version now supports FileInfo II and QMenu as well as keeping details of both the male and female trees. PC version is event driven - enter the details as they appear in documents and it generates the tree from these. Both programs easy to use with step by step tutorial. QL data and GEDCOM can be transferred to the PC version.

SBASIC SuperBASIC Reference Manual £40
Updates £6 each, £10 for 2 (Current Version - Rel 3)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on a standard QL, plus the keywords under SMSQ/E, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.

This book is ideal for all QL users and is kept up to date by regular updates.

Orders are currently being taken for the next print run of this popular tome.

(Note Price for the book does not include postage and packing).

Return To Eden v3.08	£10
Nemesis MKII v2.03	£8
The Prawn v2.01	£8
Horrorday v3.1	£8
West v2.00	£5
The Lost Kingdom of Zkul v2.01	£5

Classic QL adventures, now re-released without any need for microdrives. These include mainly text adventures, catering for all tastes, from the spoof Prawn, through to a Hammer Horror, fighting the bad-guys in the old West and battling with robotic hoards and goblins. Return to Eden is a massive three disks of adventure, with pictures for each location and a captured prince to rescue. With three characters to control, each with their own abilities and skills, this one should keep you amused for many an evening.
All six adventures are available together for only £25.

Image D v1.03 NEW!! £10

Produce graphical representations of 3D objects - view them as wireframe, hidden line and shaded. Perspective and magnification can be controlled and views can be saved to file for subsequent printing. Multiple objects can be defined and positioned relative to each other. Simple to use yet produces excellent results.

Q-Route v1.08C £25

The latest version of this popular route finding program. Find the quickest route or the shortest route between any two places, using roads. A wide range of maps is available for this program (see elsewhere in this advert). The program is easy and quick to use. You can even add your own places and roads to the maps to include local detail.

QL Cosmos v2.03 NEW VERSION £5

Ever wondered what the stars in the sky looked like 100 years ago? Or, maybe you want to learn the constellations and names of what you see in the sky. This is the program for you - generates pictures of the stars for any given place or time and provides details on these objects. Includes Halley's Comet, the Moon and the Solar System planets.

A range of games to keep you amused on the QL. Some are old favourites, like Golf and a quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for?
All 6 programs £28 only.

Open Golf v5.20	£8
QuizMaster II v2.07	£5
Stone Raider II v2.00	£5
Hoverzone v1.2	£5
Deathstrike v1.5	£5
FlightDeck v1.04	£10

These are the latest maps for Q-Route (now at v1.08C). Find your way around the various countries covered. South and West Yorkshire Map is a much more detailed area of that beautiful part of the British isles.

Britain.map v1.10	£2
BIG Britain Map (needs 2MB) v2.01	£5
South & West Yorkshire Map v1.04	£1
Ireland Map v1.00	£5
Belgium Map v1.01	£2
Catalonia Map v1.02	£2

D-Day MKII v3.04	£10
Grey Wolf v1.8	£8
War In the East MKII v1.24 (Upgrade Only)	£10

For the wargaming enthusiast - D-Day is a classic table top wargame, where you control either the Allies or the Axis forces and play against either the computer or another human player. With the ability to define your own army set ups and a choice of four different scenarios, this should keep you entertained for a while. Grey Wolf places you in charge of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

RWAP Software, 4 Anvil Crescent,
Coseley, West Midlands
WV14 8GA
TEL: 01902 836888

Cheques in £sterling
payable to 'R.Mellor'

* Also known as Trading Accounts

Are We ready for The Net?

by Dilwyn Jones

Jonathan Dent's soql project holds out the promise of full internet access from native QL hardware at last. But just how well placed are we really to take advantage of this when it arrives?

SOQL

The soql system is really the foundation for connection to the internet. It will provide the facilities required to use all those little acronyms such as TCP,IP,POP3, SLIP,SMTP and so on, even via simple BASIC commands such as OPEN. But this is just the beginning. Although Jonathan Dent has written some applications largely for his own testing purposes, these are not yet really intended to be the full applications we'll all be using to surf the net, send and receive emails etc.

So what other software is there ready for QLers to take advantage of soql?

QL-SOCKET

Users of the uQLx QDOS emulator on Linux platforms are already able to access the net, handle emails and so on, because thanks to the work of Richard Zidlicky and Jonathan Hudson, a socket library for C68 compiled programs (QL-Socket) exists, enabling access to the underlying Linux TCP/IP facilities. This gave Jonathan Hudson the head start to produce a range of applications which are currently being adapted to work with soql, so these big applications should be available almost from day one.

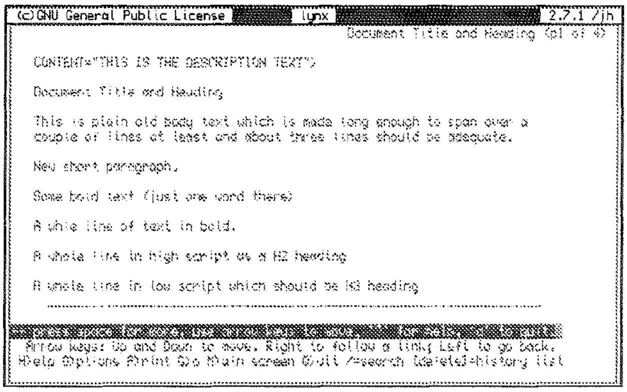
The big question of course is whether soql and ql-socket applications will be compatible with each other. Jonathan Hudson as the author of the main applications is in contact with both Richard Zidlicky concerning QL-Socket and Jonathan Dent concerning soql. To that end, Jonathan Hudson has placed this project in a facility called sourceforge.net: I quote from his web site (the quotation taken on 4th November):

"The project to provide an industry standard TCP/IP socket library based on Jonathan Dent's SOQL TCP/IP stack is hosted on sourceforge.net (http://qdos-inet.sourceforge.net)

The current package provides a very alpha socket library, qlmailer, an SMTP mailer and QPOP3, a pointer environment POP3 program. The package on sourceforge.net provides basic internet access for _native_ QDOS systems."

LYNX ported by Jonathan Hudson

This is a browser program capable of accessing the internet and browsing pages offline. It was originally intended as a text only browser. It comes in two QL flavours - a version only capable of offline browsing (version 2.3.7), rather limited but less demanding on the hardware (can run on Trump Card). The later version 2.8.2 can access the net with a suitable socket library, but is a little more demanding in terms of processing power and memory, so requires Gold Card minimum. This version provides a 'fork' capability to allow an UNGIF viewer to be used for viewing GIF picture files. It is over 2MB in length in case you decide to download it from the net at peak telephone costs time! Freeware. The extraction and installation of Lynx can be a rather frustrating and time consuming exercice, partly because of the large number of files (the sources, documentation and binaries are all in one complex zip archive). Jonathan Hudson has obviously zipped this version on a platform which does not suffer from the same limitations on maximum filename lengths as the QL filing system (36 characters) - you get various error messages warning that filenames are too long and being truncated when using QL Info-Unzip, then prompted to overwrite filenames with names which have been truncated to shorter names. A less than ideal situation for new users like myself. In the end, I had to resort to using a 'ready made' version copied from someone else's system as I could not understand all the Unix-type terminology used sufficiently to sort out why version 2.8.2 would not start up properly on my system once I had worked out the minimal set of files. If we are to seriously use Lynx as our main QL internet browser I think it would be really helpful if someone with experience of using it to write and article about setting it up. Lynx is not one of those all singing all dancing modern browsers which will do anything probably including making the coffee, it is designed as a mainly text browser, so although you'll be able to read the content of most web pages with it, don't expect anything other than the most basic of presentation of content.



```
(c)GNU General Public License          lynx                    2.7.1 /jh
                                   Document Title and Heading (p1 of 4)

  CONTENT="THIS IS THE DESCRIPTION TEXT">

  Document Title and Heading

  This is plain old body text which is made long enough to span over a
  couple of lines at least and about three lines should be adequate.

  New short paragraph.

  Some bold text (just one word there)

  A while line of text in bold.

  A whole line in high script as a H2 heading

  A whole line in low script which should be H3 heading

  Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

QDOSMAIL ported by Jonathan Hudson

This is an email package for QDOS. Uses SMTP (Simple Mail Transfer Protocol) to send and POP3 (Post Office Protocol) to receive. Freeware.

QPOP3 ported by Jonathan Hudson

A full pointer driven POP3 email client for QDOS. The package includes the binaries, sources and documentation. Freeware.

FTP ported by Jonathan Hudson

File Transfer Protocol. Freeware software used for uploading and downloading files to and from the net. One common use is for sending your web pages to your service provider's web site. Another use is for accessing ftp sites for downloading programs from the internet - the ProWesS software from PROGS in Belgium is available from their anonymous (i.e. doesn't need a password) FTP site. qlftp v0.03 provides 'reget' and .netrc (whatever those terms mean), but currently only for uQLx users with QL-Socket.

ProWesS READER from PROGS

This is a HTML page display program supplied as part of the freely distributable ProWesS windowing system. It allows the usual hyperlinks and can display text in various sizes and even display GIF files. I don't think it is yet able to display the PNG (Portable Network Graphics) and JPEG (Joint Photographic Experts Group) graphics files becoming more common on the net. Now wouldn't it be nice if the ProWesS reader could interface to soql and become a second, possibly more graphical, browser for the QL, using the ProWesS scaleable fonts etc?

PARAGRAPH by Francois Lanciault

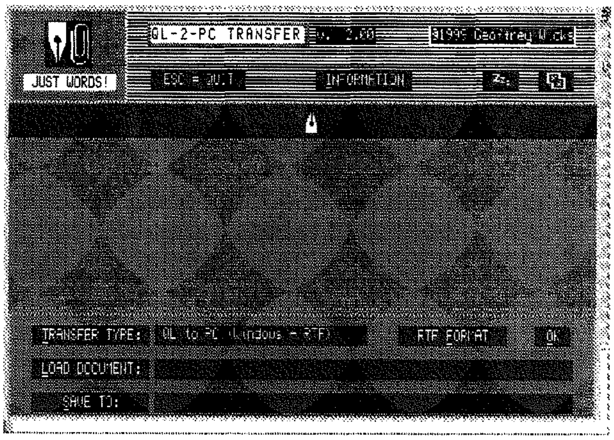
This ProWesS based commerical word processor from Canada has the facility to create HTML output in the version 2 releases - I don't yet know if this extends to full page composition or just saving the word processor's pages in the nearest equivalent HTML format.

TEXTIDY by Dave Walker

QL to PC text file transfer aid (Shareware). The most recent versions of this package also have the ability to generate a basic HTML file output.

QL2PC TRANSFER by Geoff Wicks

Just Words have included the facility to generate simple HTML file output from QL word processor files in this commercial QL to PC file transfer aid.



QL-2-PC TRANSFER v. 2.00 ©1995 Geoffrey Wicks

TRANSFER TYPE: QL to PC (Windows - RTF)

LOAD DOCUMENT:

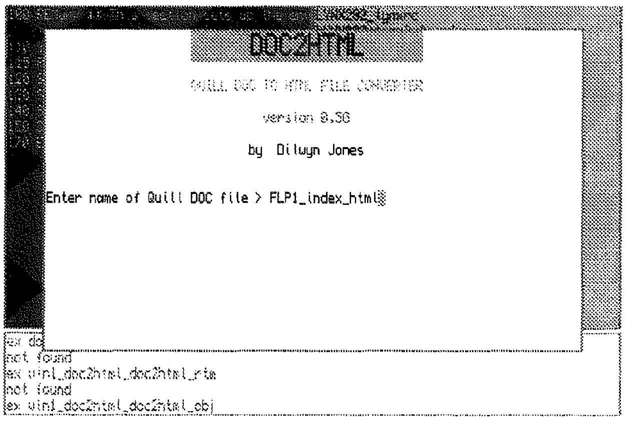
SAVE TO:

TOHTML and UNHTML by Adrian Ives

A pair of filter programs with C source code. As the names imply, for converting between text files and html files. Brief instructions and quite easy to use.

DOC2HTML by Dilwyn Jones

Yours truly wrote this freeware program for converting QL Quill files to HTML pages. Although at a basic level able to convert DOC pages into broadly equivalent HTML pages, it also has options to convert the Quill attributes (bold, underline etc) to generate links to other pages, links to graphics and so on. Not the most feature-packed of packages, nonetheless this useful little utility will enable you to produce quick and simple web pages including simple picture content without having to learn to use a new program, although you won't be able to preview its output unless you have a suitable HTML viewer.



```
DOC2HTML

QUILL DOC TO HTML FILE CONVERTER

version 0.30

by  Dilwyn Jones

Enter name of Quill DOC file > FLP1_index_html
```

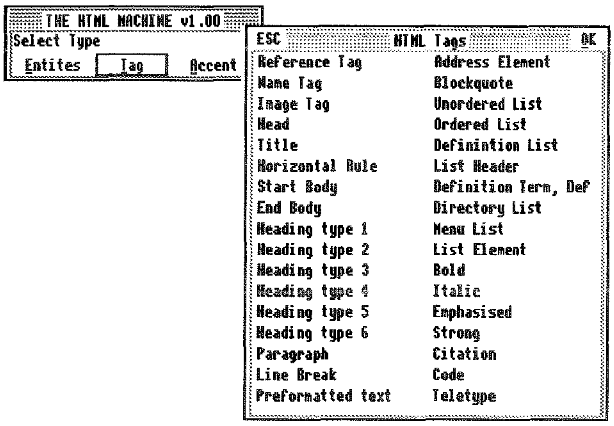
```
ex do
not found
ex win1_doc2html_doc2html_rtm
not found
ex win1_doc2html_doc2html_obj
```

TEXT2HTML by Dilwyn Jones

An earlier version of DOC2HTML. This earlier version simply adds the minimum HTML headers to allow a text file to be converted to an HTML file for further editing in a text editor (see listing below).

HTML MACHINE by Roy Wood

For those who prefer to create their web pages by hand, manually editing HTML (HyperText Markup Language) files, this freeware popup utility acts as an aide-memoire - to save you having to look up or remember the HTML tags used for the various styles etc, this program contains most of the common HTML tags in a table - simply point to the tag you need and the program places the relevant code in the Stuffer Buffer, ready for you to recover it into the destination program with an ALT-SPACE keypress. Obviously, this program needs pointer environment and the Menu Extension (QMenu). Roy also wrote an article about the HTML markup language in an earlier QL Today.



HTML TUTORIALby Norman Dunbar

Anticipating that people may want to swot up on HTML in preparation for the time when the QL hits the net in a big way (not to mention helping himself to learn HTML to create his own web site) Norman Dunbar has prepared a series of tutorials on HTML available from his Web site: www.dunbar.cwc.net/qdos/qdos.html

PRINTER DRIVERS

Several people have come up with the idea of using programmable printer drivers (e.g. PRINTER_DAT for Quill) to output text as html. All that is needed is to change the preamble printer codes to insert the Html header and Body tags for an html page, and the postamble to insert the End Body and End Html tag for an html page. Bold, for example, can be output by simply changing the normal printer control codes for bold on to ‹BOLD› and bold off to ‹/BOLD›

A very basic HTML file would look like this: ‹HTML› ‹HEAD›‹/HEAD› ‹BODY› ‹/BODY› ‹/HTML›

So the most basic of programs to convert a text file called TEST_TXT to an HTML file called TEST_HTM could look like this:

```
100 REMark Text To Html
110 OPEN_NEW #4,FLP1_TEST_HTM
120 PRINT #4,'‹HTML›‹HEAD›‹/HEAD›‹BODY›'
130 OPEN_IN #3,'FLP1_TEST_TXT'
140 REPeat read_text
150   IF EOF(#3):EXIT read_text
160   INPUT #3,text$
170   PRINT #4,text$
180 END REPeat read_text
190 PRINT #4,'‹/BODY›‹/HTML›'
200 CLOSE #3 : CLOSE #4
```

Other potentially useful programs as far as the internet is concerned are graphics conversion utilities, especially now that the long awaited "colour drivers" are available in SMSQ/E. Jonathan Hudson (is there no end to this man's talent?) has a program called Unpic on his web site which will aid the conversion of pictures between QDOS and other platforms such as PCs. Common graphics file formats on the PC include:

GIF	Graphics Interchange Format, from Compuserve
PNG	Portable Network Graphics
JPEG	Joint Photographic Experts Group
BMP	Windows Bitmap format
TIFF	Tagged Interchange File Format
PCX	ZSoft's picture compression and expansion system

BMP and PCX formats are not very common as page display formats - the most popular being GIF, JPEG and PNG.

There are QL programs such as Dave Westbury's Photon JPEG viewer for the QL (available from my web site), several GIF viewer and conversion programs such as Open World from Ergon Development (converts a number of formats in fact) and Ungif and Engif by Carlo Delhez. Norman Dunbar wrote some programs for conversion of QL screens to Windows BMP or PCX formats (called SCR2BMP and SCR2PCX respectively - available from my website and from various software libraries.

As yet, I know of no 'movie' viewers for the QL to handle animations such as animated GIFs and MPEG files, but surely this is only a matter of time.

The rather limited sound facilities (i.e. BEEP) on the QL at first made me think that we are on to a non-starter when it comes to sound files on the internet. However, the Q40 and some emulators such as QDOS Classic Amiga have the facility to play sounds via the computer's native sound hardware. This obviously got Jonathan Hudson interested, as he recently produced two programs, Sox and SoundToy, for converting sound file formats in QDOS and playing them back where supported by the hardware. As an example, these programs can convert between the Windows WAV file formats and the UB (Unsigned Byte) format used by the Q40.

To get on the internet in the first place with a QL you will need the soql system, a web browser such as Lynx, and an email program. These already exist and are being bolted together as I write this. You will also need a lot of patience and stamina I am sure, not just because the software is new, but because the whole field will be new to many QLers and there will perhaps be a steep learning curve, especially as many of the free applications were ported from software based on other operating systems such as Linux/Unix. Still, we have long wanted colour drivers, internet access etc and they are one by one all becoming reality.

THE CONTENT

And where does the QL stand from the point of view of a presence on the web? Surprisingly, there's quite a lot of material out there, including a lot of information and programs to download. Most if not all of the QL traders have their own web sites, along with many of the well known software and hardware designers such as Dave Walker (of C68 fame), Nasta, Thierry Godefroy, the entire QL Today editorial team, Jonathan Hudson and a host of others. A typical search for QL or Sinclair QL in most search engines on the web will probably return more results than you'll have time to read.

My conclusions? We are moving more slowly in the direction of the internet, world wide web and email than I'd like and the software is not yet as advanced (but perhaps less complex as a result) as the software available for Windows, Mac and Unix environments. However, "colour drivers" were something we feared would never see the light of day, but are now available on a variety of QL platforms. SOQL and QL-Socket libraries exist (the former in an early version only) and the applications are being bolted onto them as I write so that we will from day one have at least fairly basic applications, the rest is then up to the software developers. The QL-related Web content is already out there and the experience of creating and providing the content already exists among our traders and authors. Be prepared for a steep learning curve at first if you have little experience of the net and this type of software!

QL 2000

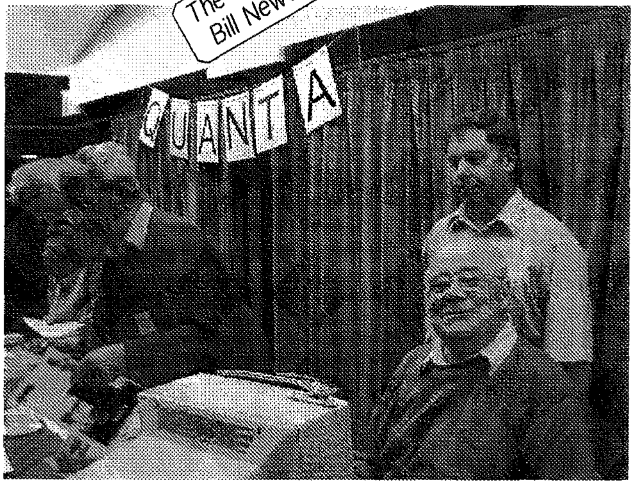
Bruce Nicholls

QL 2000 was held on Saturday the 14th of October at the Horizon Centre in Portsmouth in the UK. The show, which was hosted by Quanta, was billed as the most important meeting for the QL in recent years and on the second day of the Show an Open Forum entitled "The Way Forward" was held.

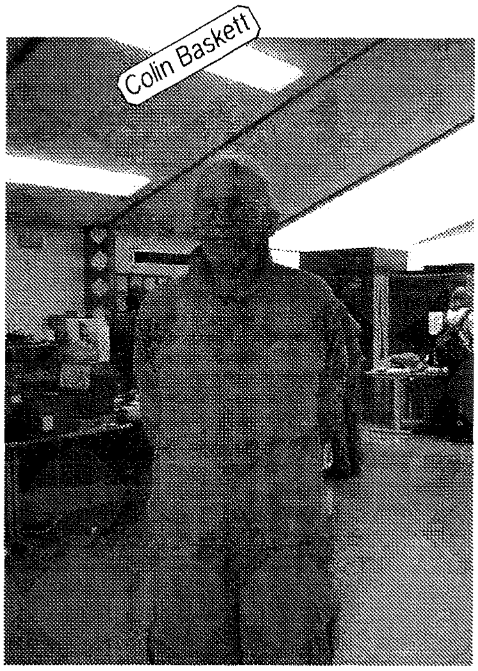
The Horizon Centre has been used for QL shows before, although not in recent times, and is easy to access from the main motorways. The centre is an ideal venue with a large hall and several side rooms for demonstrations and was fully 'professionally' catered for the whole show.

QL 2000 was billed to be the best QL show this year and I must say it did live up to being the best QL show for many years, in my opinion, for various reasons. The first was the attendance, this was like being at a Northampton show of yesteryear again with people mulling all around. The second reason was the wide diversity of the people attending for a UK show, we had most countries where the QL is still active in attendance, France, USA, Germany, Switzerland and others I have probably missed were in attendance. The last reason that this show was important was putting people in touch with one another over future projects for the QL to keep it alive.

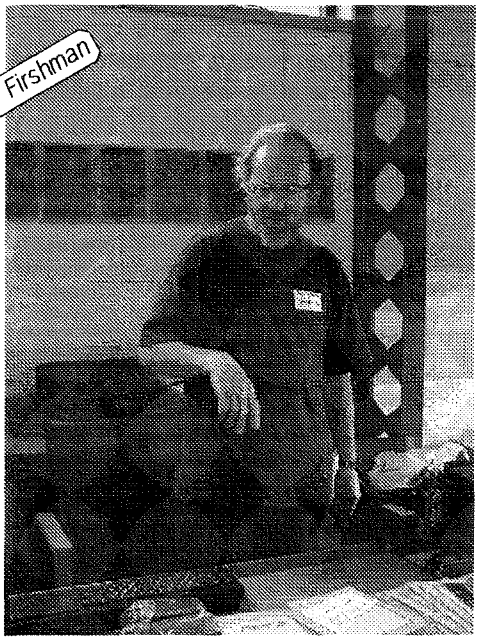
The Quanta Stand: John Taylor, Bill Newell and Roy Brereton

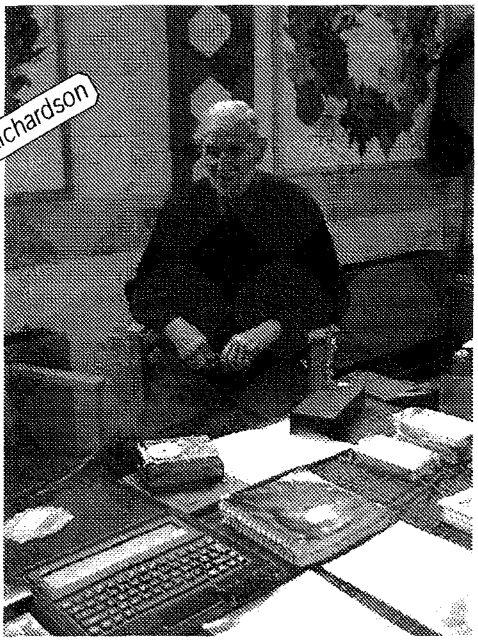


The show opened on Saturday at 10am with all the major QL traders present or represented, TF Services, Qbranch, Jochen Merz, Geoff Wicks, EEC (Bill Richardson), RWAP and Darren Branagh. Quanta had a full stand ably manned by the commitee members throughout the day. A bring and buy stand was set up outside the main hall and various other QL systems were setup around the centre.


Colin Baskett

Several new products and developments were being seen for the first time at the show and these included the Quanta CD Rom being produced by Darren Branagh on behalf of Quanta. This is a CD with a single QXL file which contains all of the Quanta Library disks. This is being sold via Quanta as the library can only be supplied to members. Darren was also selling a new Religion CD Rom containing various texts of religious material.

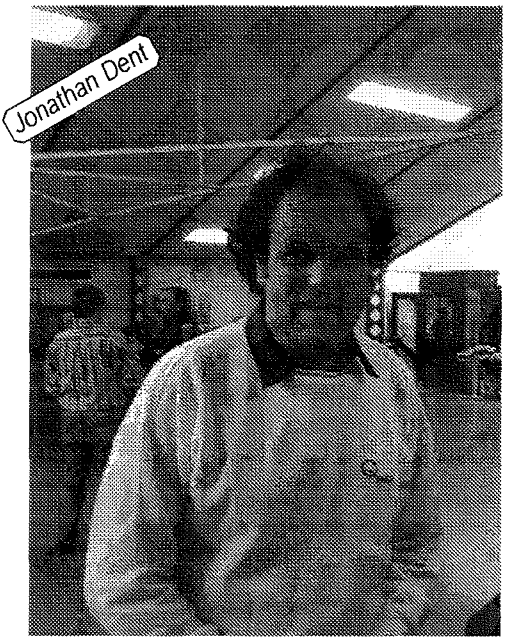

Tony Firshman

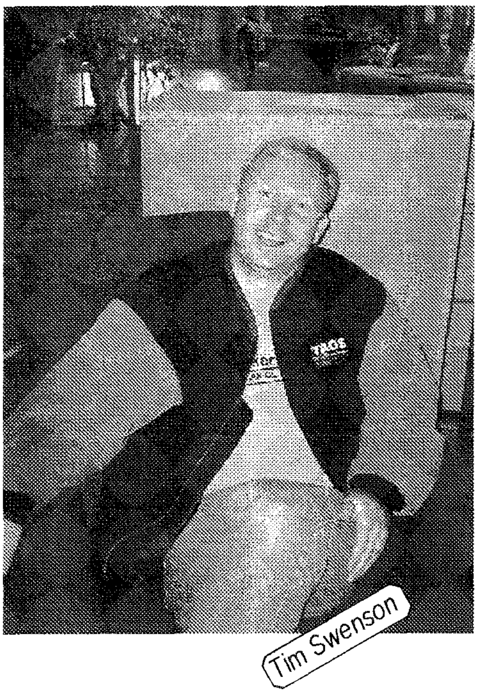

Bill Richardson

Al Boehm provides a full review of the contents of this new CD Rom on page 24 and Darren provides an insight into how to produce a CD ROM for the QL with particular reference to the Quanta Library this issue. Darren was also selling various computer bits and pieces such as 'old' 486 computers which are ideal for a QXL card. Dilwyn Jones was also present helping Darren out by producing more CD Roms due to the high demand at the show and also collecting information for this issue of QL Today.

Bill Richardson was looking in fine shape after his major operation. Bill was selling mostly z88 equipment but he did have a few ED disk drives for sale. Bill was also promoting a resurrected z88 users club magazine which has merged with a well known spectrum magazine, although the name escapes me at the moment.

Sarah Gilpin


Jonathan Dent



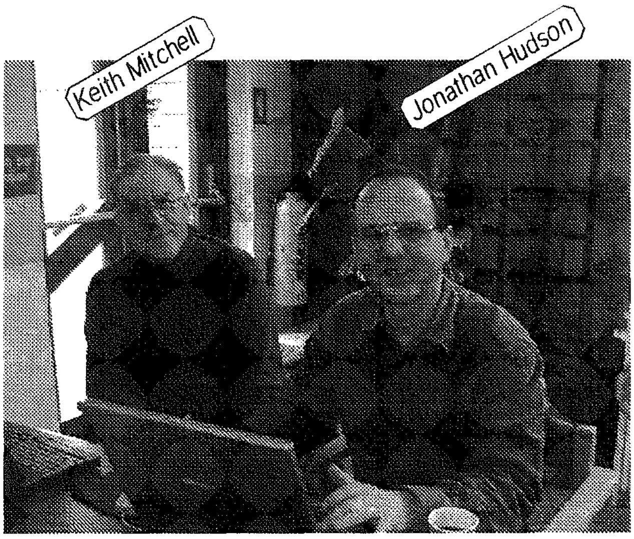
Tony Firshman was selling his usual range of QL hardware accessories together with the new Compswitch which is a 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off. This is a must if you use any of the latest printers or zip drives as they now come with no on/off switch.

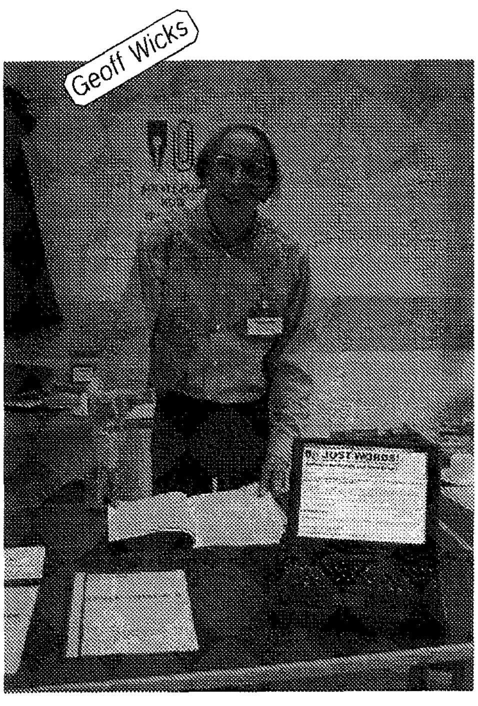
Geoff Wicks was demonstrating his range of software including QL2PC word processing translator software. He also started his new range of 'poundware' software at the show, more details are given in the news section of this issue.

Rich Mellor of RWAP software was unable to attend the show due to ill health but Sarah Gilpin kindly agreed to represent Richard at the show. Image D, a program to allow the creation of three dimentional views of objects, is the latest software to resurface under the RWAP banner.

Qbranch aka Roy Wood was selling his large software selection together with Q40's and other hardware items. Roy together with Jochen Merz was also giving out the special present for QLToday readers in the shape of a mouse mat with an inbuilt currency converter, a nice momento of the show.

Jochen Merz was present with the all new QPC2 V2 pre-release software taking centre stage. In this version of the software Marcel has been able to create a full 'windowed' version of QPC2. This means QPC2 no longer has to take up all the screen area when running as it can be resized and left in a window on it's own with other programs open around it. The other major feature of this release is the inclusion of the Colour drivers. Suqcess author Wolfgang Uhlig was also demonstrating a new version of the software in which the sorting of fields within a database can easily be changed just by clicking on the field name.



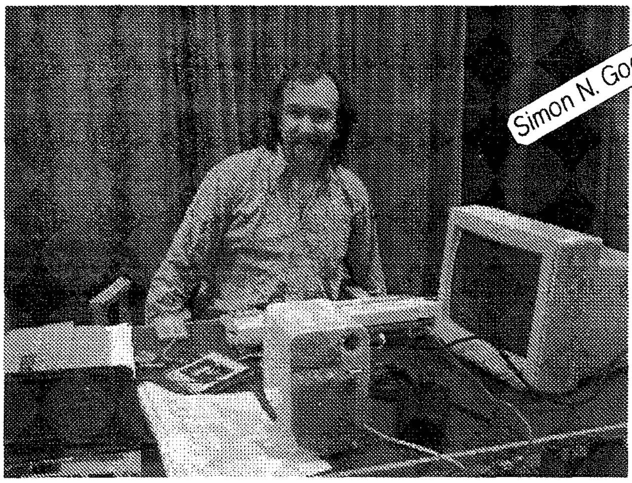


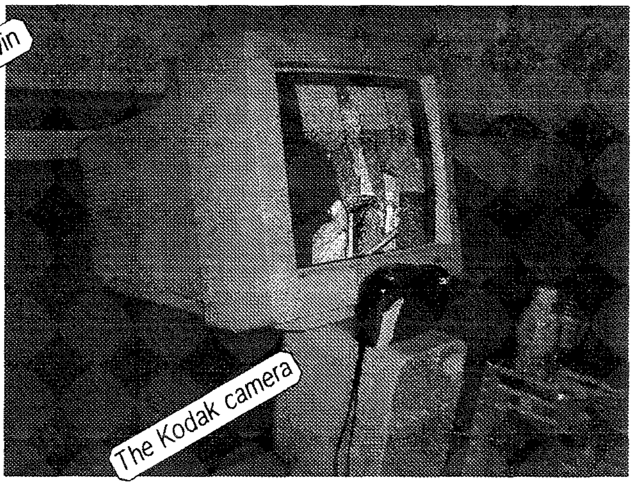
Saturday was a trading day but it was also a time to put faces to the names of people you may have only read about. Tim Swenson of QL Hacker's journal was present from the USA. Tim launched the first Q40/Linux Journal at the show and had the latest QL Hacker's journal number 33 in hardcopy form, contact swensont·lanset.com for more information. AL Boheim demonstrated the QL midi interface by Simon Goodwin and Simon also demonstrated his new sound device which allows sound files to be copied to a normal QL device driver to produce very good quality sound. Stuart Honeyball of Miracle systems also made a brief appearance.

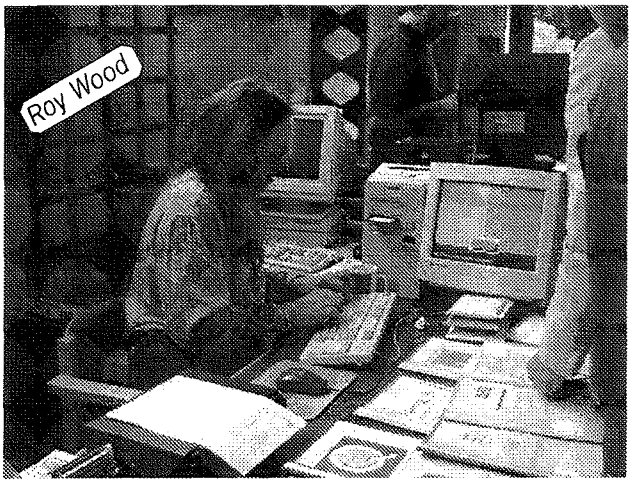
Jonathan Dent who has produced the first native QL TCP/IP system was also present giving out trial copies of his SOQL system, he was also introduced to Jonathan Hudson who got straight to work in fully testing the system, we hope to see further developments soon on the QL TCP/IP front.

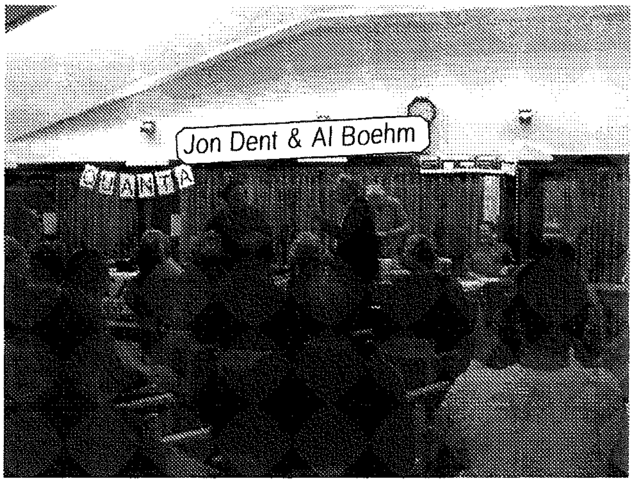
Saturday finished with the grand Quanta Convoy through Portsmouth to get to a restaurant for the evening meal.

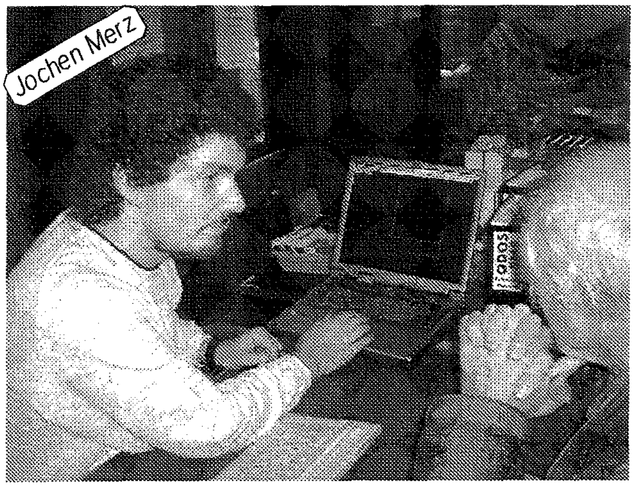
In the next issue an account of the Sunday together with the Quanta open forum "The Way Forward".


Simon N. Goodwin

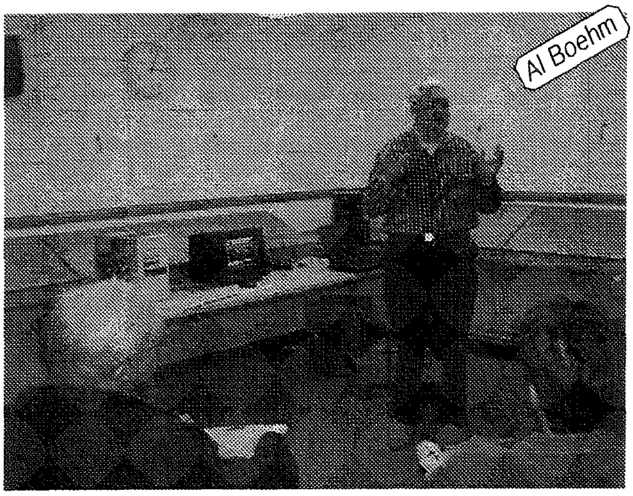

The Kodak camera


Roy Wood


Jon Dent & Al Boehm


Jochen Merz

Robin Barker


Al Boehm

Dilwyn Jones

Small ads

Very rare, but it is possible - and it is cheap! In case you are interested to place a small ad in QL Today: up to 50 words cost only DM 5,- / EUR 2.56, up to 100 words cost only DM 10,- / EUR 5.12. We will try to accept ads even after the deadline, but we can't promise that it will catch the current issue if you are late. Please indicate exactly which text should go in, whether all/part of your address, phone number etc. should be printed or not!

QL 2000 - The American View

Tim Swenson

After 13 years of QLing, I've finally been to a Quanta sponsered QL show. Granted I've been to 6 Sinclair/QL shows in the US, but the QL 2000 show was my first European show. I think I'm safe in saying I was also the person who traveled the longest distance to get to the QL 2000 show (about 5300 miles).

Thanks first go to Roy Brereton for picking my family and I from Gatwick and driving us to Portsmouth. It took a bit to get used to be a passenger on the "driver's" side of the car. By the end of my trip I could not get used to all of the roundabouts. Once at the hotel, my wife decided that being up about 23 hours was enough and took a nap. I wanted to adjust as quick as possible so I stayed up. Colin Baskett was nice enough to take myself and the Boehm's on a local site seeing trip. We visited a 2,000 year old Roman palace and a 1,000 year old Saxon church. In the US to get history this old, we have to import it. Later that evening after being up about 30 hours, I finally got to sleep.

The morning of the show was cool, brisk, and a little bit damp. It reminded me a lot of the North Coast of California. I arrived just as the show was starting. The usual traders that I had met before were there; Tony Firshman, Roy Wood, Bill Richardson and Jochen Merz. The other traders were familiar to me, but I had never met them before. I quickly introduced myself to Darran Branagh of Q-Celt. Sitting with him was someone that I have long known via e-mail, Dilwyn Jones. Geoff Wicks was enough to spot as he has a good picture on his web page. On the Quanta table I was introduced to the folks there. Most of the names were familiar enough. There was one vendor I did not know or meet. A rather older gentleman with a full white beard was selling a variety of older computer items. Almost forgot, I did meet the folks from one of the Quanta subgroups with the table just outside the main hall (oh, to remember names).

I did not spend much of the first day looking over the vendor tables. I did not travel all that distance just to buy what I could get via mail order. I spend most of the time meeting and greeting the varous QLers. I met Bruce Nicholls for the first time. Like the others, we had known each other for a number of years, just by e-mail.

My lunch break was filled with fiddling with English coins and trying English foods. The quickest way to solve the coin problem was to show the cashier what coins I had and let her pick out the ones that come to the right amount. By the end of my trip, I had a good feel for what each coin looked like. My first taste of English food was a Pork Pie, something I had heard about from watching BBC shows. It was different.

The highlight of the day was seeing Simon Goodwin, Chris Lyle, and their son, Ingo. The previous year Simon and Chris had stayed at my house for the West Coast Sinclair show. At that point, Chris was just pregnant (but did not know it). At the same time, my wife was about 8 months pregnant with our third. The show was the first time that my wife and I had seen Ingo and that Simon and Chris had seen Megan.

Later that evening, after much discussion about directions, we all arrived in downtown Portsmouth at the Post House for dinner. I don't remember the gentlemans name that shared our table, but I do remember his Yorkshire accent. After a few days I was getting familiar with the different English accents.

The next day came soon enough and back to the Venue for more QL stuff. Now it was time to meet more QLers that I had heard about. Jonathan Hudson looked younger that I the picture on his web page. Just by accident I happened to sit down next to George Gwilt for lunch. When he mentioned "his" assembler to somebody else, I started to get a clue on who he was. For Sunday's lunch I tried the a-typical English Steak and Kidney Pie. It was not too bad. Later I met Arnould Nazarian, a big proponent of SMSQ/E and Stella.

Soon enough the show came to a close. I had to say goodbye to all of the folks that only the day before I got to meet face-to-face. I doubt I'll get another chance to come to another QL show in Europe, but at least I made this one. Now I'm back to keeping in contact with the QL world via e-mail and web pages.

BYTS OF WOOD

SAW POINTS OFFCUTS AND SNIPPETS

October was a hectic month. Quite apart from the effort at the end of September at getting the column together I had to prepare for a rash of shows.

This year saw my first Italian show and I enjoyed meeting the Italian users very much. Not only was the food in Regio Emilia very good but the trip itself was very enjoyable even if Tony Firshman did manage to navigate us off the autostrada just before the airport turnoff so we had to double back and pay the toll all over again.

After this we had the QL 2000 show in Portsmouth. I know there will be a definitive coverage of this event elsewhere in the magazine because I saw Bruce taking copious notes but there is one aspect of this that I would like to pursue a little further.

Before I do I would like to apologise to the French QL people for my non-attendance at the show in Paris. As I said last issue, my wife is about to give birth to our second child. Like all good computer products the baby is, as I write this, about two weeks over the due date but I felt that it would not be a good idea for me to be absent from home immediately prior to the time when it was due to be born. I had to cancel my plans to attend the Paris show and, since it had still not arrived, Eindhoven and the Stafford All Formats fair as well. Normal service will be resumed in the New Year. She must be here by then - I hope.

For those of you with new diaries I hope to have the next Sussex User Group Workshop in late February - probably the last Sunday. See the back page of the next issue for details.

And now read on......

Source For The Goose

During the discussion on Sunday about the future of the QL John Taylor brought up my suggestion that Quanta offered to buy the non-colour versions of SMSQ/E from Tony Tebby so that it could give it away free to any members who wanted it. This would have the double advantage of moving most of the user base up to a more modern operating system and, at the same time, giving TT a sum of money which would buy his time towards finishing the colour drivers on all platforms and the other updates that people would like to have.

Simon Goodwin objected to this on the grounds that he, and people who use emulators on systems such as the Amiga, would get little benefit from the investment. This is, I suppose, a fair comment although I do suspect that the number of users this would effect is minimal. The other objection was that SMSQ/E was not 'open source software'. This, in my view, is a thornier issue.

I have no objection to the concept of open source software and I have a great admiration for people who decide to give their hard work away for free but we are not all identical units and our motivations and ideals are not all the same. Tony Tebby makes his living by writing software and, if we are to ask him to devote some of that time to producing software for us we must expect to recompense him. For people like Simon, Jonathan Hudson, Theirry Godefroy etc. programming is obviously something they want to do for free - this is their choice and I won't try to force money into their hands if they respect the right of others to charge for their services. Demanding that everyone makes their programs, and the source code for them, free and open may well drive away the few commercial programmers that we have and we need them now more than ever.

Programming is, however, more than just the mechanical bolting together of code to get a finished product. It is more of an art form than most non-programmers can recognise. It is also true that every programmer has a signature, a way of performing tasks which is unique to him, an this signature may be something that they wish to retain as their own. I respect that wish.

Pumping Code

Of course Simon brought up the argument that he would only use the JS ROM code that has been made open source and developed by Mark Swift to run on both the Amiga and the Q 40 and that this was the way forward because, since the code was available to all, anyone could add to it and develop the system further. Good concept except that, apart from Mark's sterling work on the Q 40, I have seen no further development. Maybe it is out there and I have not seen it but those QL users who do not use either an Amiga or SMSQ/E are still stuck with the same buggy JS or JM ROMS that they bought 15 or so years ago - so no change there then.

On the other hand those people who invested in SMSQ/E have event handling, high resolution screens, multiple executable SBASICS, extended keyboard tables and, on some systems, 16 bit colour.

No contest as far as I am concerned.

And Greek Goddesses

I have not mentioned Minerva here - as Tony Firshman would be quick to point out if I did not remedy the situation by adding a few words to the mix. Laurence Reeves put in an appearance at the QL 2000 show and said that was going to make Minerva open source too. This does effectively give people a platform to work from which is better than the older ROMs but this still lags behind some of the developments which have put SMSQ/E into the forefront. This is not to say that any of the systems are inferior in essence only that they have not had the advantage of sustained work and development in recent years. Some people have dismissed some of this work as mere tinkering with the code and I have had people complaining that is it 'no longer compatible with QDOS'. To that I suppose that we must hold up our hands and say that, yes, in some areas it is no longer compatible with QDOS but that is usually because QDOS was either inadequate or flawed. True there have been other bugs appearing but they do get mopped up when reported.

So much of what is said when we get onto the subject of operating systems and machines is some kind of tautology. The 'my teams better than your team' divisive behaviour. There are some things in Minerva which I think are truly innovative but it still lacks a lot of the useful features that I find in SMSQ/E. I missed the 'compose characters' function in Minerva when I migrated to SMSQ/E because it was so intuitive compared with the laborious task of remembering all of the key combinations that produce these on a normal keyboard system. I will, however, stick with SMSQ/E for my QL work because I prefer its extended features and that is the prime consideration. Use not Ideology.

Future Use

One other aspect of the discussion at the Portsmouth show was the future of the QL. Some people felt that it had no future since it had been overtaken in many aspects by the PC but that does not take into account the ease with which anyone can produce short programs which can perform real tasks on the QL and its derivatives. SuperBasic is a very flexible and functional language and many of the people who still use the QL have written their own procedures for things which no programs exist either for the QL or for other platforms.

However much one may dislike Archive (and I always hated it) it also had a superb programming language which became the basis for the programming languages of the PSION Organiser II and Series 3. One of my customers imports books from Africa and has a whole mass of programs written under Archive to control the stock and to produce invoices. These programs are customised to this particular business. To get someone to write this for the PC would cost a fortune and to reproduce it on a PC would involve buying a separate program, 'Visual Basic', because the PC does not have an in-built programming suite.

As I have said on many occasions in the column, there is no reason why you should not use the two at the same time and, with QPC, in the same box. Another of my customers recently bought a PC and a copy of QPC 2. Having no PC-experience he called on his son to give him a hand installing and setting the system up. When he got to installing QPC 2 his son became very interested. After he had been shown what his father did with it and what it could do he called me and purchased a copy of QPC 2 for himself. Maybe this is what we should look into - to introduce the system to new users.

Retro Use

There is a web site called the Register which announces computer oriented news and is, quite often, very scathing about the big players. One of its bulletins recently concerned a large computer distributor who, during a clearout of its warehouse, discovered a large cache of unopened Sinclair ZX 81s. Of course these are not as rare as the Z80, which can fetch a high price in good condition, but it did provoke a great deal of interest.

There is a general feeling of nostalgia for the older systems and some of these change hands for large sums of money. Old TV games such as PONG or PAC man are fetching good prices at the auctions and 8 bit computing is generally looked on with a lot a lot of fondness.

An article in a recent 'Observer' weekend magazine about computer games mentioned how the bulk of today's games programmers were self taught on the old Spectrum and other machines of that era. They started of with a bit of basic got hooked on programming and moved on. Without that easy entry level to the world of programming many of them would not have got started and things would have been a lot different. I do not expect the QL ever to become a work force in the computing world but it would be a good introduction to the joys of coding.

Boot Hill

An interesting thread on the QL Users internet group has been the re-emergence of the concept of a 'program installer'. This is an idea that Steve Hall and I kicked around for a while a couple of years ago and is based on the way that many of today's systems use an installation program to add new programs to the system. This assumes that you have at least a hard disk or RomDisq to use as a basic system and then copies the appropriate files to specific locations making changes to the system files loaded as it goes. Windoze relies on just such as system for its functionality, and is conspicuously fragile because of this.

This does not mean that it is a bad idea. Many people have problems in setting up a system and many others have migrated to other systems, notably Windoze, because of the ease of new product installation. Of course there is a school of thought which says that users should at the very least write their own boot files and wherever possible re-compile/assemble the code (see above). Many users, on the other hand, want a program that works - somewhat understandably I would hazard. Again there is nothing wrong with people experimenting with code and that is all very well but there is a functional side to a computer that should not be overlooked. I support this idea but it does require a degree of conformity that we, as QL users, have previously avoided. It is something that should be looked into and should be discussed. Do let us know what you think. Drop QL Today a letter and we could have a new forum for the discussion.

As a footnote to this I would like to point to two experiences which really point to the way we use our systems and the way in which we see them. Both Simon Goodwin and Tony Tebby have used my system in the past and both of them made assumptions about the way I had set it up. They both pressed 'ALTKEYS' and waved the mouse around with great rapidity - and then wondered why it did not do what they expected (in Simon's case he managed to call up Duncan Neithercuts ClipScrapBoard and then lock it up - something I have never managed to do). Amazingly enough this is something that Windoze Users have been able to do for ages. Not because Windoze is a better system but because the system arrives pre-configured and most users are too lazy (or don't know how) to change the default settings. This said I basically support the idea and I would like to see some serious research done into the means to get it organised.

That's ProGress

Marcel Kilgus recently remarked on the user group newslist that there was something odd about the ProWesS fonts under the new colour drivers. Joachim van Auwera who wrote the ProWesS system has been away wearing his other hat as a Triathlon athlete for most of the summer but Marcel's comments and the work he put in trying to modify some of the drivers in the current release prompted him to look into it and he found a few modifications he could make to the code to get rid of the annoying '&wait' statements in the startup file as well as a few other improvements to the way it runs and looks. It is well worth downloading the latest release from the PROGS website and upgrading your system.

Those of you who have upgraded to the newer colour drivers have to upgrade to the latest version of ProWesS if they want to use it with the new drivers. If you do not do this then the display will be unreadable. The best way to go about doing this is to download the whole new ProWesS installation from the website and then use the 'Install Software' followed by the 'Update Installation' options from the ProWesS Utilities menu. Once you have done this you will have to check that the line:

```
D mode33_pfd
```

is added to the Proforma_cfg file in the pws_mine_ subdirectory. This loads the drivers for the new colour mode and allows ProWesS displays to be rendered in a way that makes sense. You should also ensure that the QVME_pfd driver is the latest one since that could also cause display problems.

Its just Norton Old Boy

I got a call from one of the long standing QL users this month who was puzzled by the behaviour of some of his code under QPC 2. He wrote a short piece of basic to test out a couple of things in his new setup having just moved from QPC 1 on a 486 laptop to a shiny new Windoze 98 machine. when he ran the program it looked for files that he had forgotten to put on the floppy and he got an instant lockup. He could find no reason for this since it did not act the same on his other machine, a desktop, or his Qubide based system.

The answer lies in the preinstalled software provided by the suppliers. Most PC stores are so convinced that the end user is a congenital idiot who is barely capable of writing his his name on the credit card slip that they pre-install a whole slew of stuff on the machine before they ship it. In this case he had a copy of the Norton Antivirus program installed and that was causing the problem. It intercepted a call to a device that was not present and intercepted this as virus behaviour but, for some reason, did not latch into its normal behaviour which would be to pop up a window offering to quarantine the offending code. Maybe this is because it is not running in the normal Windows mode and you cannot see the options. All I know is that it is wise to leave a formatted floppy disk in the drive in order to avoid these problems.

Seasonal Offerings

As we move into the festive season I thought that you might be interested in a few of the Christmas Shows available to computer fanatics. Here is a selection:

- Serverella Ali Chipset and the Forty Bugs
- Parsing Boots
- Jack and the Directory Tree
- Toad of Load All
- The Linux, The Glitch and the Hard Drive
- Tales of Dot-Matrix Potter
- Little Read Instruction Book
- Goldilocks and the Three Bear Drives

featuring all of your favourite characters, Barren Hardisk, Buttons (he was framed), Floppy the bunny, and Disk Whittington.

There will be the usual rash of films:

- The Bios in the Hood
- The Cruel 'C'
- SuperBasic Instinct
- The Forth Man (a sequel)
- Silence of the LANs
- The Windows of DOS
- The Soundcard of Music
- Hex Files and Microdrive Tape

and of course a whole rash of Bond movies featuring Q Branch.

On the TV there will be a Christmas special by that Queen of daytime TV, Oprah Win3_ as well as a documentary on fraudulent Spiritualism called 'Bad or Changed Mediums'.

And Music...

Don't forget to look out for the new record by Kiki Dee's younger sister Dee V. Dee - 'No Windows No Cry', A Jimi Henrix re-release 'All around the Midi Tower' and The Pointer Sisters singing the theme song from the 'Sticky Mouse Club'.

What ever you do over the Festive season have a good time and I'll see you all in 2001.

WISHLIST TO SANTA QLAUS

WE HAVE ASKED FOR YOUR WISHES SOME TIME AGO AND WE WOULD LIKE TO SEE IF THE DEMANDS HAVE CHANGED OVER THE YEARS! TELL US WHAT YOU WOULD LIKE TO SEE!

Happy QLing New Year!

NEXT ISSUE

You have just read through another jam-packed issue of QL Today! This time, several articles did not fit into the current issue because we simply ran out of pages! To make sure we can provide you with more interesting issues, write to us! Everything is welcome! The better the mix, the more attractive it is for our readers - including yourself!
The next issue will carry on with the existing series (Assembler programming, GoldFire, Programming ProWesS in BASIC, to list a few). Dilwyn will start an interesting article about expansion possibilities for your QL - also aimed at people who have not touched their QL for a while and re-joined the QL community. Jérôme Grimbert explains his excellent new sprite editor in details.

We would also like to present you a list of upcoming shows - show organisers, please inform us!